修士論文

GAN を用いた遮蔽された人体骨格情報の再構成

18861628　張　博超

指導教員　朝香 卓也

2020 年 1 月

首都大学東京大学院 システムデザイン研究科
システムデザイン専攻
電子情報システム工学域

目次

表目次

図目次

第1章

序論

1.1 背景

近年，カメラを用いた人物検出，姿勢推定，行動認識の技術は，セキュリティ監視，自動運転，スポーツ解析などに利用が広がりつつある．駅や工場等の施設への監視カメラの設置が増加しており，安全安心見守りや監視業務の負担軽減のために，人工知能による動画や映像からの人物検出，姿勢推定，行動認識に関する技術への期待が高まっている．また，自動運転技術の発展により，障害物の位置や動きの認識，周辺の歩行者や自動車の状況の把握等に関する技術が注目されている．

一方，深層学習を用いた人体骨格情報推定技術は，カメラのみによる複数人物の骨格推定が可能であり，2次元座標として関節位置情報が推定できる．2020年オリンピック・パラリンピックの開催により，骨格情報推定技術を用いたスポーツ選手の行動認識，移動追跡等に関する研究は広く行われており，スポーツ動作解析のツールとして幅広い利用が期待される．

しかし，これら技術の中核をなす骨格情報推定技術は，遮蔽されていない部位の検出精度は高いものの，遮蔽された部位の検出が困難である．カメラによる対象者の全身の骨格を観測する時，人物大の装置，部品棚，柱，看板などの邪魔になる遮蔽物が存在する場合，対象者の関節検出と骨格情報推定が困難である．骨格情報推定技術を用いた人物検出，姿勢推定，行動認識が難しくなり，精度が低下する．

従って，骨格情報推定技術を用いた認識技術に対して，遮蔽に頑健な処理手法が非常に重要である．例えば，深層学習を用いて，人物の動きを時系列データとして学習し，遮蔽された部位の骨格情報を復元することが可能になる．再構成した骨格情報を用いて，遮蔽が発生した場合でも，人物検出，姿勢推定，行動認識が高精度を維持できると考えられる．

1.2　目的

遮蔽された部位の骨格情報を再構成するために，連続フレームの時系列骨格情報だけでなく，1 フレームあたり人体骨格情報の制約条件も考慮する必要がある．遮蔽が発生した場合は，時系列骨格情報の中で，遮蔽なしの連続フレームのデータと遮蔽ありの 1 フレームのデータを同時に考慮する必要もある．

そこで本研究では，深層学習を用いた遮蔽された部位の服装，手足，動作，表情などを復元することを最終目標とし，その重要機能の 1 つとして挙げられる遮蔽された部位の骨格情報座標データの推定手法を提案する．敵対的生成ネットワークに基づく時間的空間的生成ネットワークと識別ネットワークを構築し，図 1.1 に示すように予測技術を用いた遮蔽された部位の骨格情報座標データを生成する手法の提案を目的とする．具体的には，骨格情報技術を用いて，動画から人体骨格情報座標データを取得し，作成した生成ネットワークにより，1 フレームの骨格情報座標データを生成する．

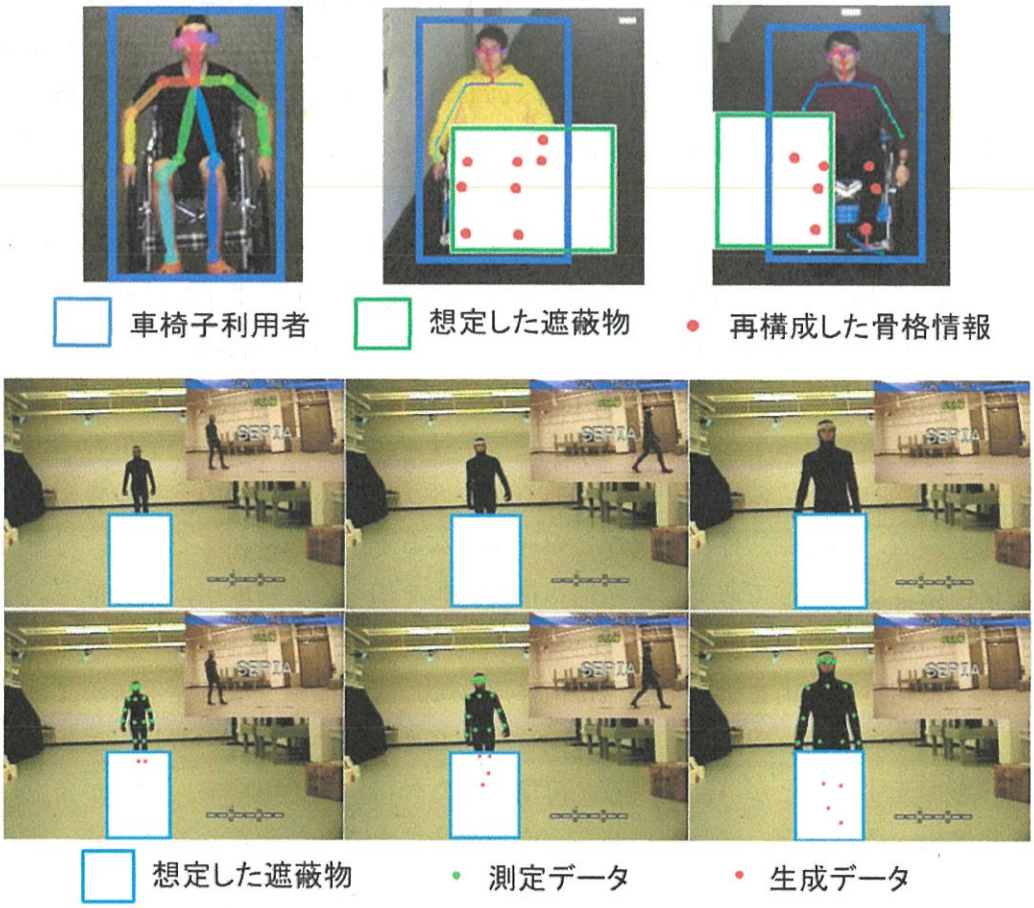


図 1.1. 遮蔽された部位の骨格情報座標データの生成のイメージ図

1.3 構成

本論文は，6 章により構成する．本章では，本研究の研究背景と目的について記述する．まず第 2 章で関連研究とその問題点について記述する．次に，第 3 章で提案手法の中で使用されたニューラルネットワークの説明について記述する．更に，第 4 章では，提案手法であり，遮蔽された人体骨格情報再構成の概要とニューラルネットワーク構造について記述する．そして，第 5 章では，シミュレーション条件，データセット，提案手法と従来のモデルの比較結果，考察について記述する．最後に，第 6 章では，本論文のまとめと今後の課題について記述する．

第 2 章
関連研究と現状

2.1 従来の遮蔽に頑健な検出手法

本節では，既存の遮蔽に頑健な対象者の検出手法について述べる．

Chen らは遮蔽ありの状況において，複数カメラを用いた姿勢推定手法を提案している [1, 2, 3, 4]．これらの手法では，Kinect カメラを用いてフレームごとのスケルトンを抽出して 25 個の身体関節の 3D 座標を取得し，座標変換を使用して 2 つ以上の Kinect によってキャプチャされるジョイント座標を共通の座標系に統合する．しかし，予めシーン内に全身骨格情報を統合できる違う観測角度を持っている複数の Kincet カメラを設置する必要があるという問題がある．

Shum らは遮蔽ありのフレームにおいて，遮蔽されていない部位を用いた姿勢推定手法を提案している [5]．Shum らが提案している手法では，追跡対象の信頼度を評価する一連の測定値を設計し，信頼性推定をモーションデータベースクエリに組み込むことにより，運動学的に有効な一連の類似した姿勢を取得する．しかし，Shum らの提案手法は，見えている部位の骨格情報から似ている姿勢を検索し，遮蔽部位を推定できるが，事前に大量の姿勢の骨格情報データを保存する必要がある．また，時間的制約条件を活用していないため，これらの手法は全身遮蔽の場合に利用できない．

Ukyo らは部位追跡の併用によって遮蔽時でも車椅子利用者を検出できる手法を提案している [6]．これらの手法では，事前に構築した検出器による車椅子利用者の検出に加え，部位ごとの追跡を併用し，追跡結果から車椅子利用者の位置を推定する．具体的に，まず，追跡対象の部位ごとに遮蔽されているか否かを判定する．次に，遮蔽されていない部位との位置関係と，過去の位置の変化に基づいて遮蔽された部位の位置追跡を推定する．これにより，遮蔽に頑健な車椅子利用者の検出を実現している．

以上に挙げた先行研究は本研究が志向する深層学習を用いた遮蔽された部位の骨格情報の再構成とは異なる．

2.2 従来の骨格情報を用いた動作予測と認識手法

本節では，機械学習による既存の骨格情報に基づく人体動作の予測と認識手法について述べる．

2.2.1 MLP(Multilayer Perceptron) を用いた動作予測方法

Horiuchi らは機械学習を用いた人体の動きの予測手法を提案している [7]．MLP とは，パーセプトロンを多層構造にした階層型ニューラルネットワークである．パーセプトロンとは，1958 年 Rosenblatt らによって提案されたニューラルネットワークの起源となるアルゴリズムである [8]．

図 2.1 に Horiuchi らが提案している手法とニューラルネットワーク構造の概要を示す．深度画像センサにより 25 個人体関節 (75 個座標データ) を検出して身体の重心を計算し，10 フレーム (0.33 秒) の時系列骨格情報データを MLP モデルに入力し，1 フレームの骨格情報データを予測する．提案している MLP モデルは 6 層構造であり，780 個特徴量を入力層に入力し，4 層中間層を通じ，出力層から 78 個座標データを生成する．

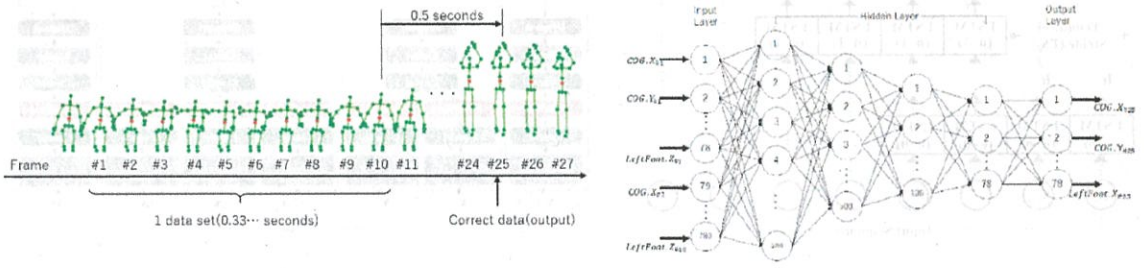


図 2.1. MLP を用いた動作予測の概要図 [7]

2.2.2 LSTM(Long-Short Term Model) を用いた動作認識方法

時系列骨格情報に基づき，Zhang らは深層学習を用いた人物の動作認識手法を提案している [9, 10]．LSTM は，入力ゲート，忘却ゲート，出力ゲートによる構成され，従来の RNN(Recurrent Neural Networks) では長期的な依存関係を学習することができなかった問題を解決したモデルである．

Zhang らの提案手法は，図 2.2 に示すように，骨格情報座標データを二次加工し，3 層の LSTM モデルに入力し，Softmax 層からスコアを融合し，動作を分類する．図 2.3 に Lee らが提案している時間スライディング LSTM を示す．Lee らの提案手法は，取得した骨格情報

座標データも二次加工したが，特徴の属性を考慮し，異なるタイムステップの LSTM に入力し，動作を分類する．

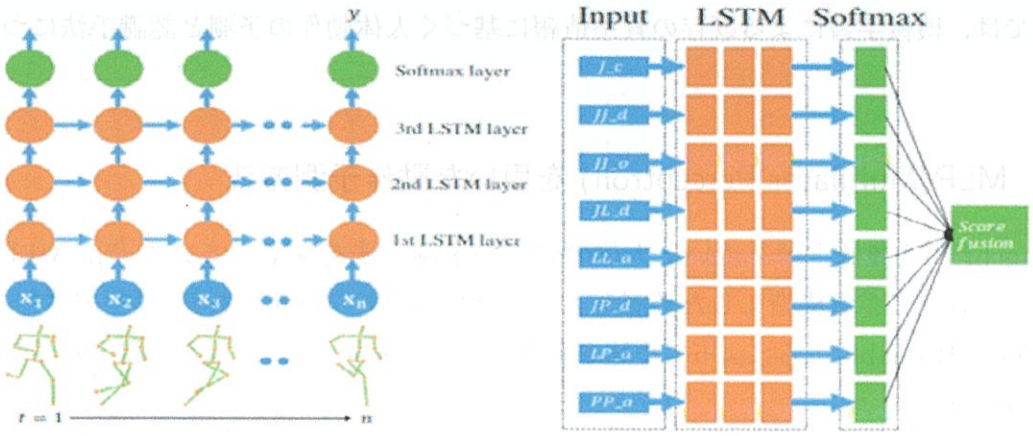


図 2.2. 多層 LSTM を用いた動作認識の概要図 [9]

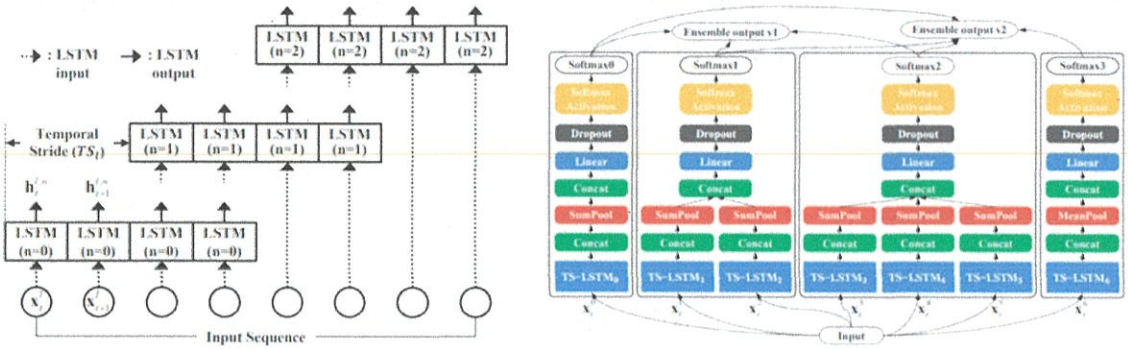


図 2.3. 時間スライディング LSTM を用いた動作認識の概要図 [10]

2.3　複合モデルを用いた時系列データの予測手法

本節では，時系列データに対し，複合機械学習モデルを用いた既存の予測手法について述べる．

2.3.1　CNN(Convolutional Neural Network)-LSTM を用いた時系列データ予測方法

Kim らは時系列データに対する CNN-LSTM を用いた予測手法を提案している [11, 12]. CNN は，順伝播型人工ディープニューラルネットワークの一種であり，入力層，畳み込み層，

プーリング層，全結合層，出力層による構成される．畳み込み層は，ノードにフィルタ処理して特徴マップを得る．プーリング層は，畳込み層から出力された特徴マップを新たな特徴マップとする．

Kim らは住宅のエネルギー消費に対する CNN-LSTM を用いた予測手法を提案している.Kim らの提案手法は，図 2.4 に示すように，多変量時系列エネルギーデータを CNN モデルに入力し，LSTM モデルを通じ，全結合層からエネルギーを予測する．図 2.5 に He らが金価格の予測に対する提案している CNN-LSTM モデルを示す．毎日の金価格を固定タイムステップに分割し，2 層の LSTM モデルに入力し，Attention モデルを通じ，CNN モデルを入力し，畳み込み層，プーリング層，全結合層から金価格を予測する．

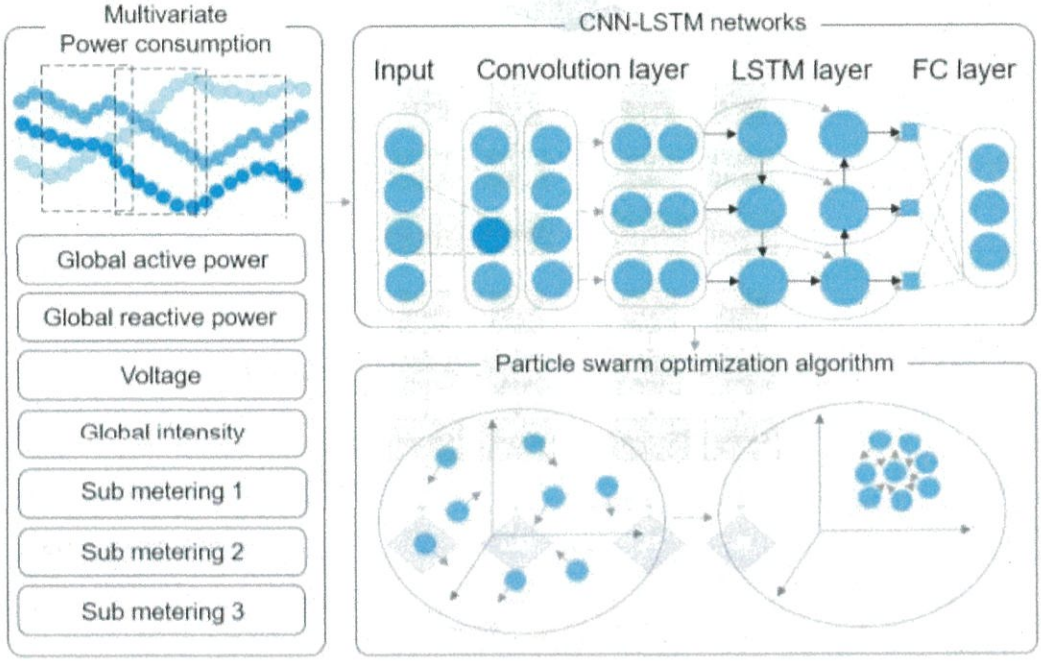


図 2.4. CNN-LSTM を用いたエネルギー消費予測の概要図 [12]

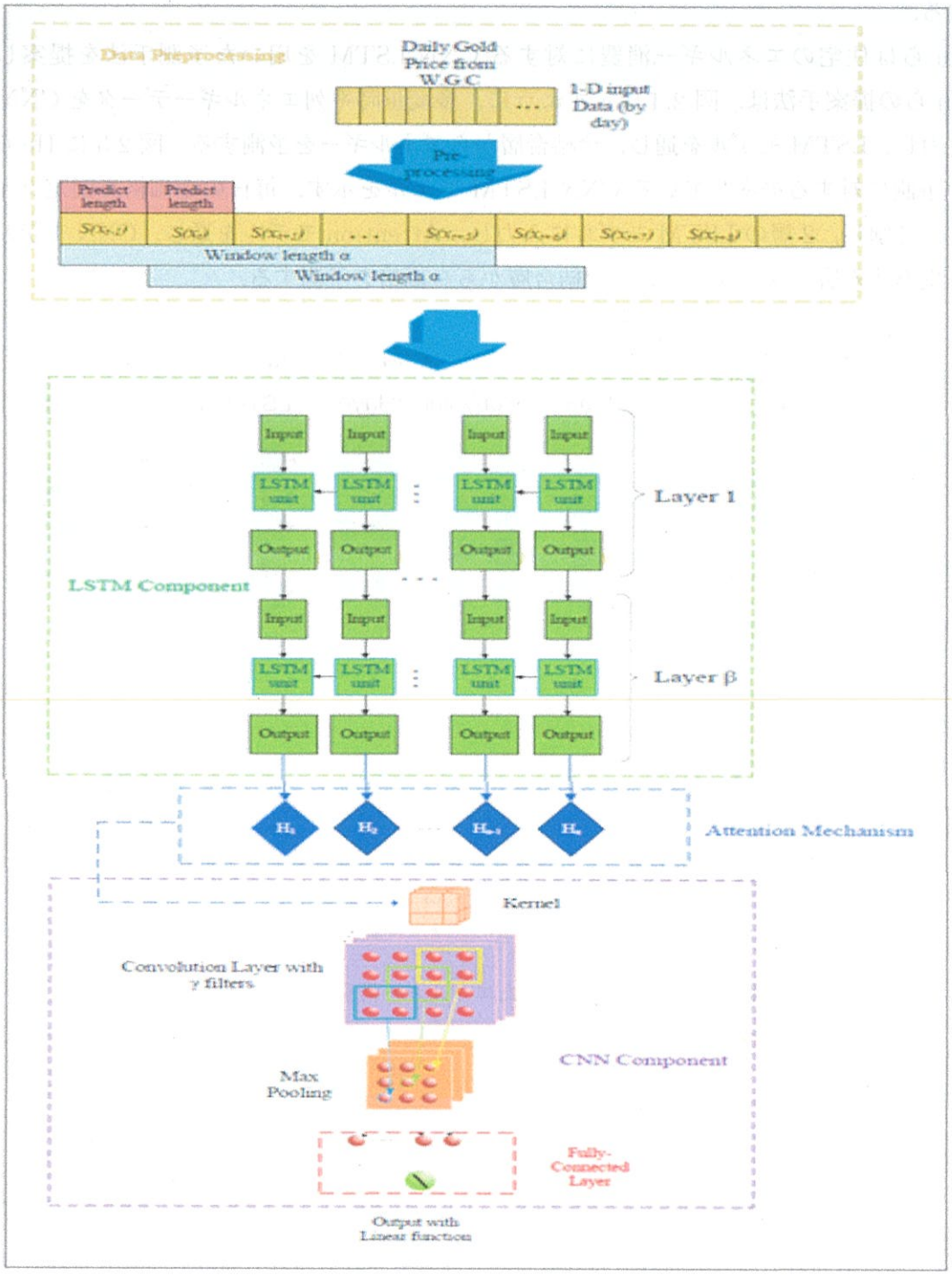


図 2.5. CNN-LSTM を用いた金価格予測の概要図 [11]

2.3.2 GRU(Gated Recurrent Unit)-GAN を用いた時系列データ予測方法

Barsoum らは GRU-GAN を用いた人体の動作予測手法を提案しているる [13]．これらの手法では，時系列関節位置と角度データを利用し，WGAN に基づく確率的な人間の動き予測のための新たなシーケンス間モデルを提案されている．

GRU は，LSTM をシンプルにしたモデルであり，入力ゲートと忘却ゲートを更新ゲートとして 1 つのゲートに統合している [14]．図 2.6 に GRU-GAN の構造を示す．Generator と Critic のトレーニングを切り替える Critic を示している．Discriminator は人間の動きの実際のシーケンスと偽のシーケンスを区別することを学習する．Generator を更新するには、一貫性の損失と骨の長さの変化に加え，WGAN-GP の損失を使用する．10 フレームの入力から 30 フレーム以上を生成できる．

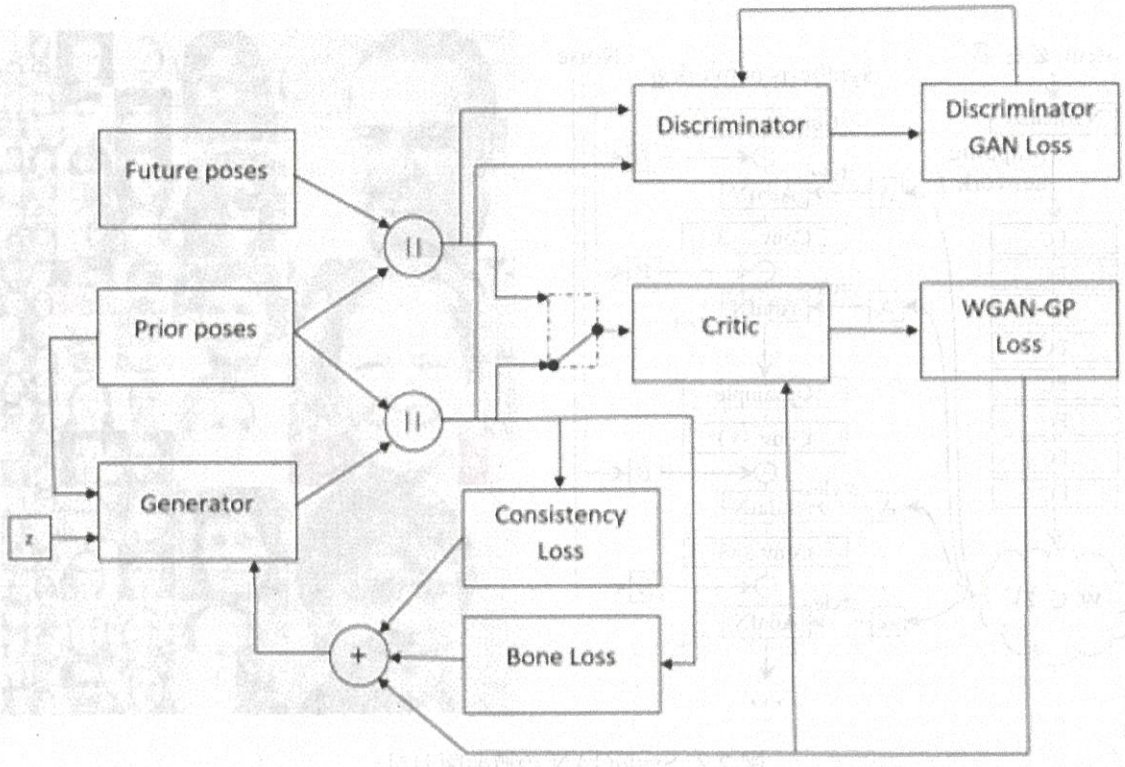


図 2.6. GRU-GAN を用いた人体の動作予測の概要図 [13]

2.4　GAN(Generative Adversarial Network) の派生モデル

2.4.1　StyleGAN モデル

StyleGAN は Karras らによる提案されている各解像度ごとの画像特徴の挿入手法である [15]．提案されているアーキテクチャにより，高レベルの属性 (ポーズとアイデンティティ) と生成された画像 (そばかすと髪) の確率的変動が自動的に学習され，教師なしで分離され，合成の直感的でスケール固有の制御を可能にする．

図 2.7 に StyleGAN の仕組みと生成した画像を示す．最初の層ではなく途中の層にスタイル A と確率的な影響 B を追加する．AdaIN(Adaptive Instance Normalization) を用いて画像生成を行うことにより，高解像度かつ自然な画像生成を可能にする．StyleGAN を用いた存在していない物を生成する研究がある [16, 17, 18]．

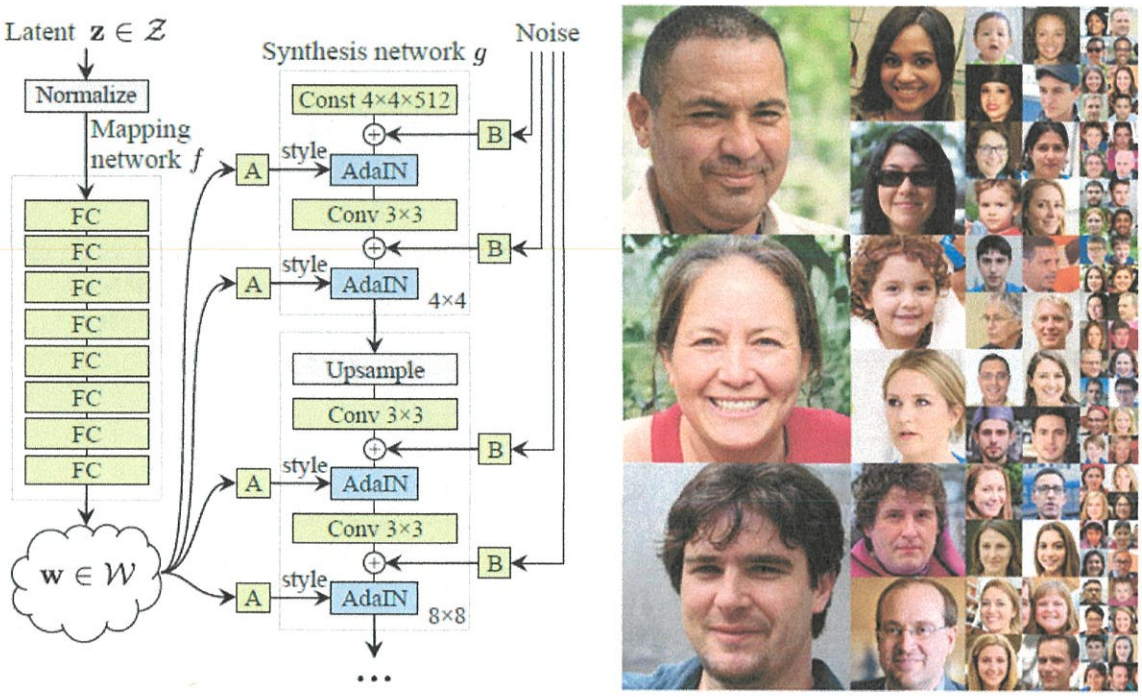


図 2.7. StyleGAN の概要図 [15]

2.4.2 TecoGAN モデル

Chu らは低解像度ビデオを高解像度に変換できる超解像 TecoGAN を提案している [19]. この研究では，空間の詳細情報を犠牲にすることなく，時間的に一貫したソリューションに導くビデオ超解像度の敵対的トレーニングを提案されている．時間的敵対的学習は，現実的で時間的に一貫した詳細情報を達成するための鍵であることを示す．

図 2.8 に TecoGAN のアーキテクチャを示す．TecoGAN では，空間的な高周波の詳細情報と時間的な関係の両方を考慮し，GAN を用いたフレーム間の動きを一致させている．生成ネットワークは低解像度の入力から高解像度のビデオフレームを生成するために使用される．一貫性のフレームを生成するために、生成したデータを再利用する．識別ネットワークでは，生成された 3 つの時間的に隣接した高画質フレーム，正解フレームと低画質フレームを判別して学習する．TecoGAN に基づく超解像技術に関する研究が存在している [20, 21].

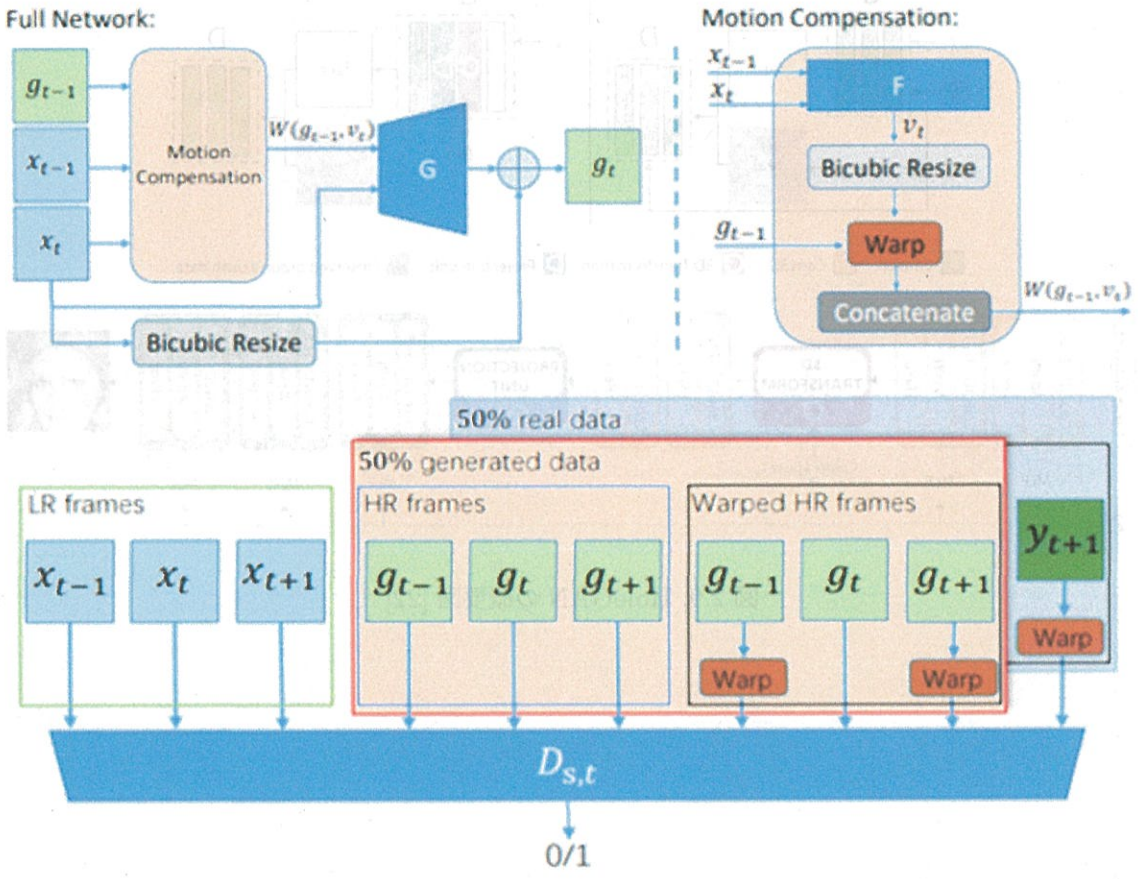


図 2.8. TecoGAN の概要図 [19]

2.4.3　HoloGAN モデル

自然画像から 3D 表現を学習する生成モデル HoloGAN は Thu らによる提案されている [22]．HoloGAN は代わりに世界の 3D 表現を学習し，この表現を現実的な方法でレンダリングする．他の GAN とは異なり，HoloGAN は，学習した 3D 機能の剛体変換により，生成されたオブジェクトのポーズを明示的に制御する．

Thu らの提案手法は，図 2.9 に示すように，ラベル無し 2D 画像から 3D 表現を学習するために，3D 畳み込みを Generator に導入する．三次元空間においてデータを生成し，剛体変換を利用し，二次元平面に投射し，最終画像を生成する．HoloGAN がラベルのない画像から形状と外観を分離することを学習し，要素を各自に操作できることが示されている．GAN を用いた自然画像から 3D 表現する他の研究もある [23, 24]．

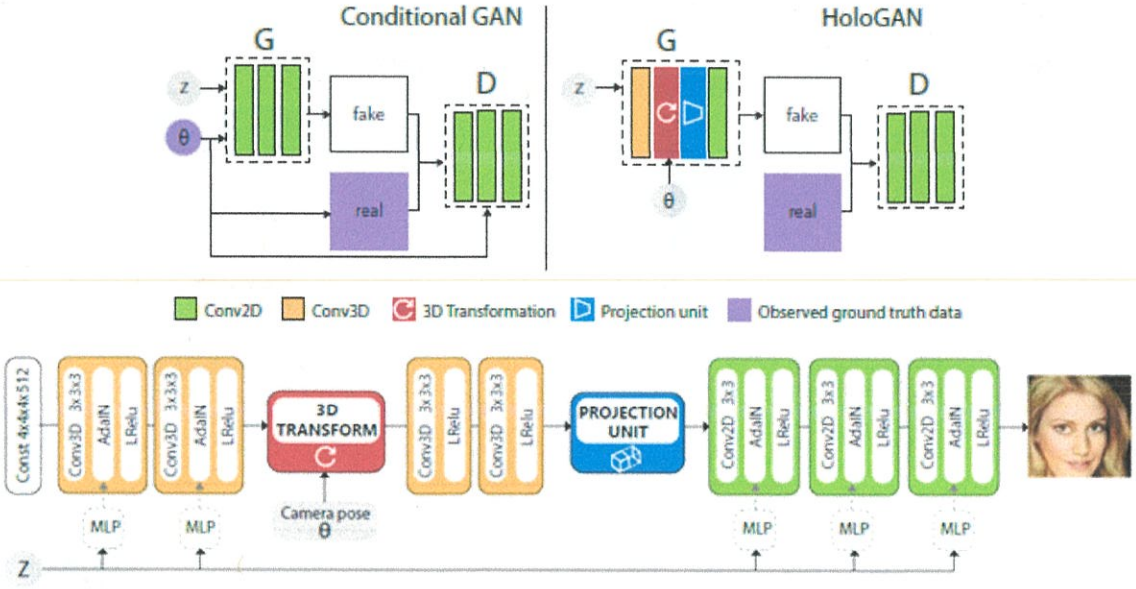


図 2.9. HoloGAN の概要図 [22]

2.4.4 CycleGAN モデル

Zhu らは画像のスタイル変換に関する CycleGAN を提案している [25]．入力画像と出力画像間のマッピングを学習することを目的とする視覚およびグラフィックの問題において，ペアのトレーニングデータは利用できなかった．Zhu らはペアの例がない場合に，ソースドメイン X からターゲットドメイン Y に画像を変換する学習方法を示す．Zhu らの目標は，$G : X- > Y$ のマッピングを学習し，$G(X)$ からの画像の分布が敵対損失を使用して分布 Y と区別できないようにすることである．

図 2.10 に CycleGAN の概要図を示す．CycleGAN は，2 つのデータソース間の変換を学習する GAN の一種であり，単一の変換を多段階の変換に分解することにより，画像の変換品質を高めるだけでなく，粗い画像から高い画像への画像変換を可能にする．前の段階の情報を適切に活用するため，現在の段階の出力と前の段階の出力の動的な統合を学習するための適応型融合ブロックが考案されている．CycleGAN を用いた画像のスタイル変換に関する研究が多く存在している [26, 27, 28, 29, 30]．

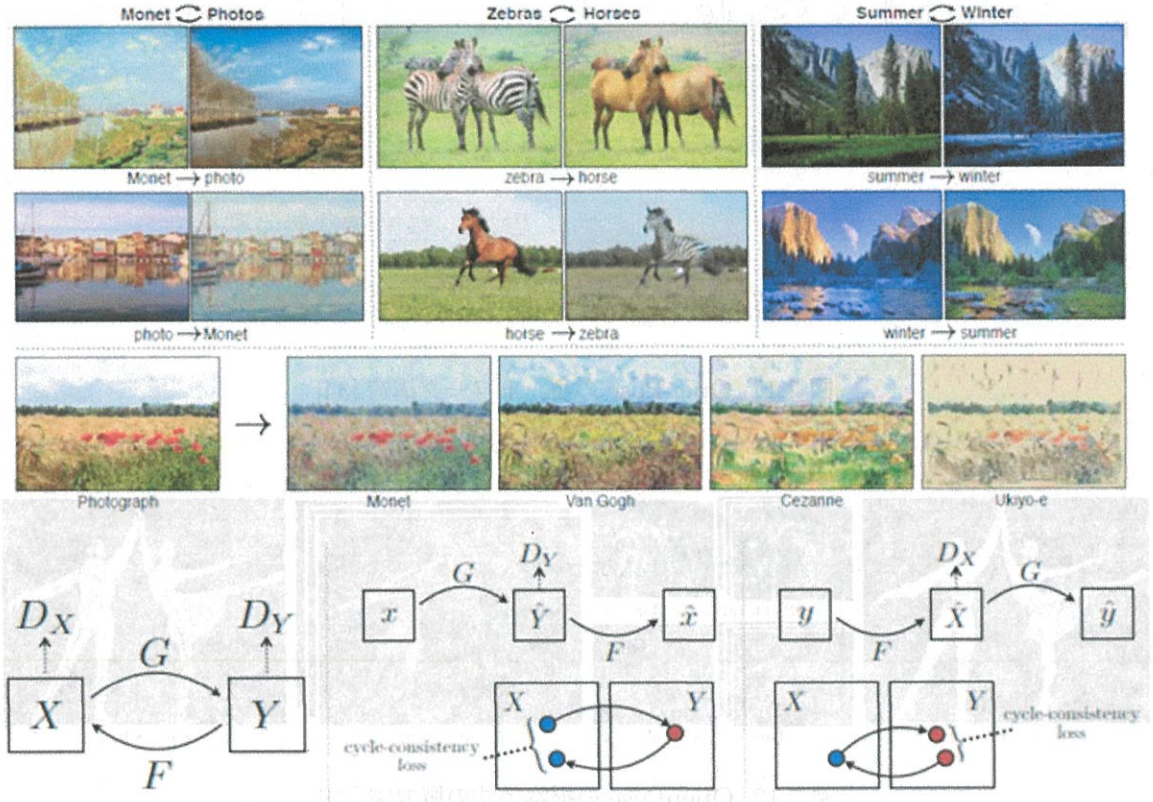


図 2.10. CycleGAN の概要図 [25]

2.5　骨格情報推定技術

近年，深層学習の発展に伴い，Kinect を上回る骨格情報推定技術が次々に登場した [31, 32, 33, 34, 35, 36]．OpenPose はカーネギーメロン大学の Zhe Cao らによって 2017 年に公開された深層学習を用いた骨格推定技術である [34]．複数人の骨格情報を単眼カメラのみによる高精度かつリアルタイムに推定でき，ライブラリとして無料公開されている．

OpenPose は図 2.11 に示すように 2 分岐マルチステージ CNN のアーキテクチャである．OpenPose の提案手法では，図 2.12 に示すように入力画像から部位の位置の推定 (S,Part Confidence Map) と部位の連関を表す 2D ベクトル (L,PAF) を算出し，S と L の集合から同じ人物の部位を組み合わせ，姿勢の状態を出力する．OpenPose では，画像中における解析対象の鼻や肘，膝などのキーポイントと呼ばれる 18 個特徴点の 36 個座標データ，及び各特徴点の推定信頼度を出力できる (図 2.13)．

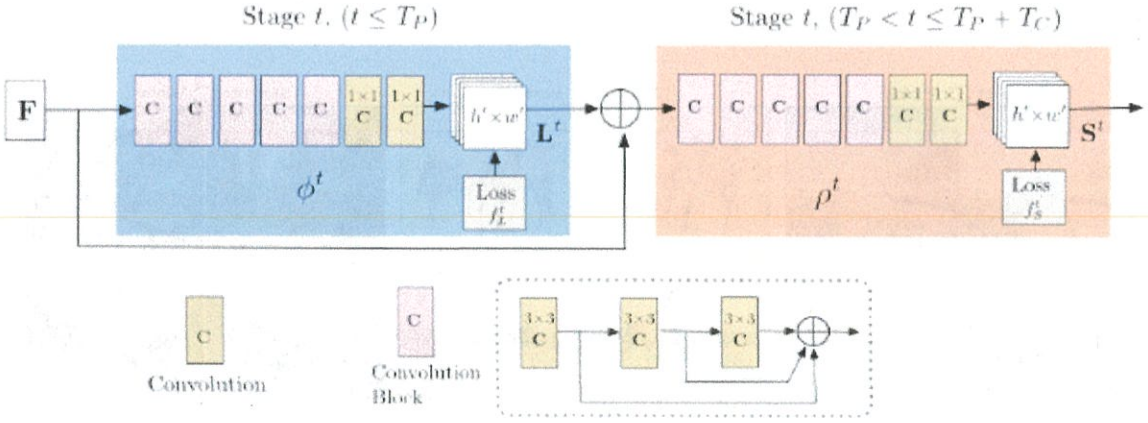


図 2.11. OpenPose のネットワーク構造 [34]

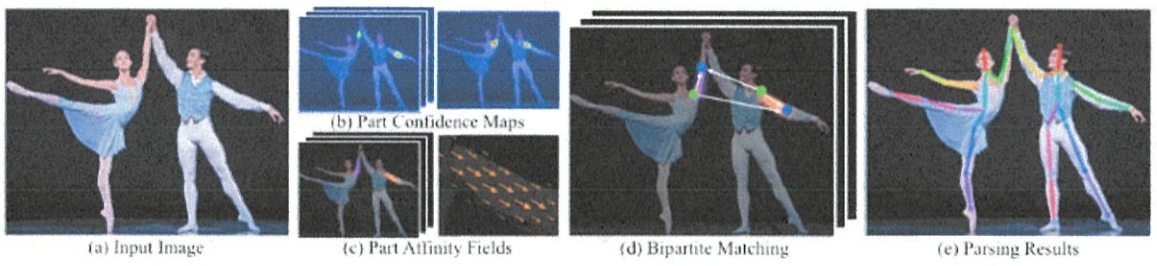


図 2.12. OpenPose の提案手法の概要図 [34]

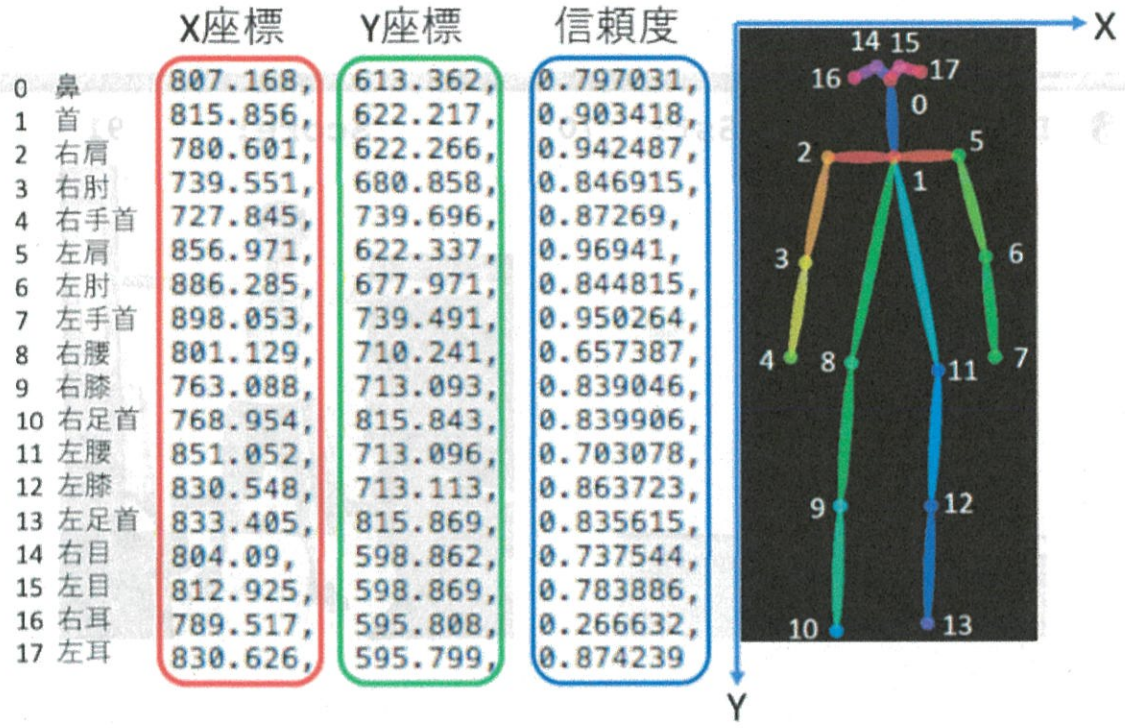


図 2.13. OpenPose による取得したキーポイントの例

骨格情報推定技術の進歩に伴い，OpenPose による骨格情報を用いた発展研究が盛り上がっている．スポーツ分野における応用例として，OpenPose を用いたジェスチャー採点システムという研究がある [37]．この研究では，人の関節の 2D 位置と身体のスケルトンワイヤフレームをキャプチャし，すべての関節の運動軌跡の方程式を計算する．図 2.14 に示すように，変更可能なスコアリング式を使用し，姿勢を採点する．また，泳者のスイムストロークを分析する研究が存在している [38]．この研究では，図 2.15 に示すように，水中で撮影した泳者の映像から OpenPose による骨格情報を取得し，SVM と Random Forest モデルによる関節角度と関節座標データを用いた姿勢を識別する．更に，OpenPose ライブラリの拡張により，手と顔の推定が追加し，単一の RGB フレームを使用するリアルタイム手の 3D ポーズ推定の研究がある [39]．この研究では，図 2.16 に示すように，単一の汎用 RGB カメラを使用し，OpenPose を用いた手の関節の 2D 位置を推定し，手の 3D モデルが推定された 2D 関節位置に適合し，手の 3D ポーズが復元される．

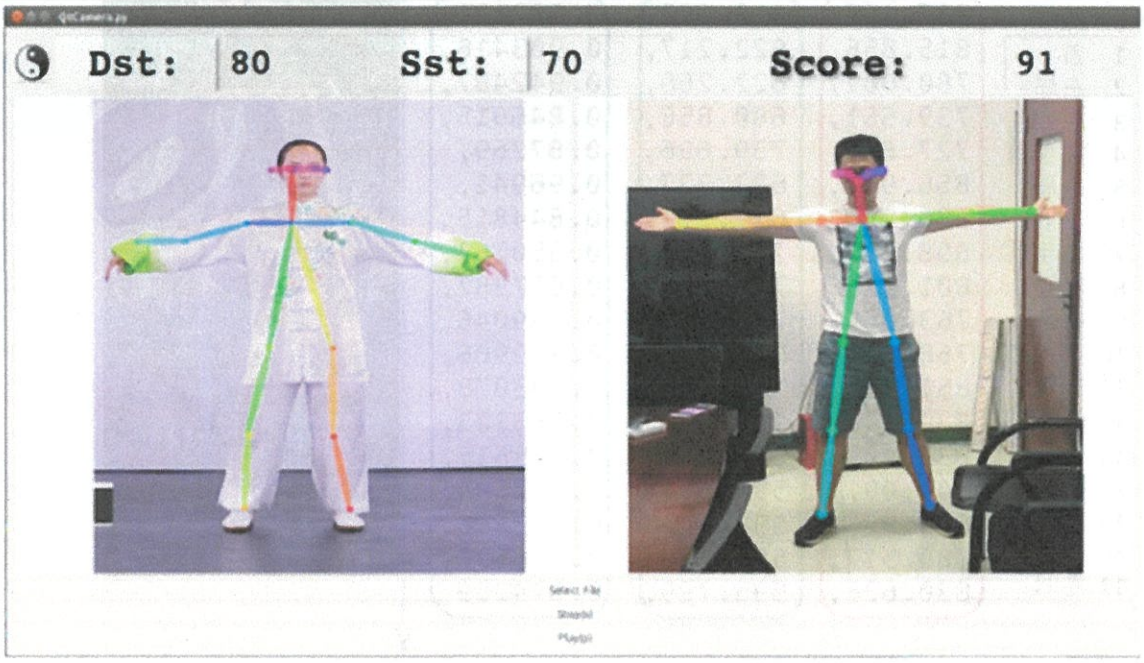


図 2.14. OpenPose を用いたジェスチャー採点システム [37]

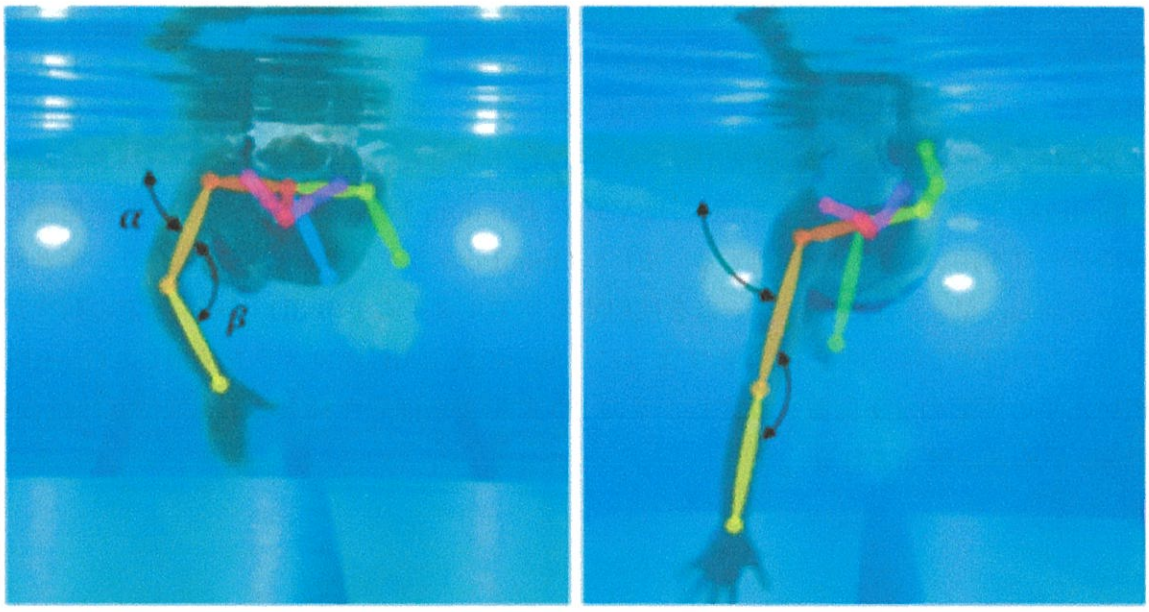


図 2.15. OpenPose を用いたスイムストローク分析 [38]

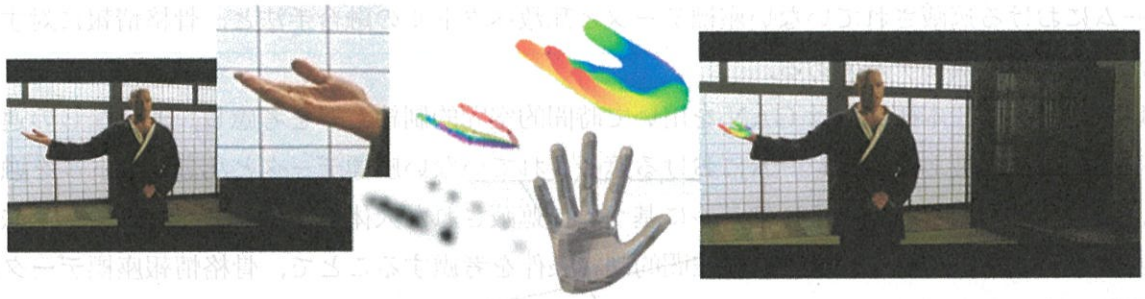


図 2.16. OpenPose を用いた手の推定イメージ図 [39]

2.6　本研究の位置づけと方針

2.1 節において，遮蔽された部位の骨格情報を推定する手法について概観した．少なくとも 2 個以上の違う観測角度の複数カメラを利用することにより，取得した関節座標データを統合し，高精度に遮蔽された部位の骨格情報を推定できるが，予めシーン内に全身骨格情報を統合できる違う観測角度という位置関係を決定し，複数 Kincet カメラを設置する必要があるという問題がある．また，遮蔽ありのフレームに対し，遮蔽されていない部位の骨格情報を用い，データベースに保存している骨格情報とマッチングし，似ている姿勢を推定できる一方で，保存していない姿勢に対して骨格情報の推定が困難であり，事前に大量の姿勢の骨格情報データを保存する必要があるという問題がある．

2.2 節と 2.3 節において概観したように，1 つのカメラのみを利用し，Deep Learing による時系列骨格情報を予測する手法が多く提案されている．従来の提案手法において時間的制約条件のみを考慮し，単一または複合モデルを用いた時系列データを予測する．しかし，これらの手法では，連続フレームの骨格情報を利用して次のフレームの骨格情報を予測できるが，部分遮蔽の場合に，遮蔽されていない部位の骨格情報という空間的制約条件を考慮していなかった．

2.4 節では，GAN の派生モデルをまとめた．GAN と CGAN の概要は 3.1 節で説明する．GAN の派生モデルから GAN は Generator と Discriminator の 2 つが競合する技法であり，制約条件とランダムノイズを入力すると，ランダムサンプル画像だけではなく，スタイル変換や画像詳細の再構成を制御できる画像を生成できることが見える．しかし，GAN の派生モデルでは，いずれも画像を直接の入力とするものであり，本研究が志向する骨格情報座標データの生成手法とは異なる．

従来の骨格情報予測技術では，MLP や LSTM や GAN 等の深層学習による連続フレームの骨格情報を利用して連続フレームの骨格情報を予測できるが，本研究との違いは遮蔽ありのフ

レームにおける遮蔽されていない座標データと乱数ベクトルの融合手法と，骨格情報に対する新たな複合の予測モデルである．

以上を踏まえ，本研究では GAN を用いて時間的空間的制約条件を考慮し，遮蔽なしの連続フレームと，遮蔽ありのフレームにおける遮蔽されていない座標データと乱数ベクトルを融合し，LSTM，Attention，MLP モデルに基づいて遮蔽された人体骨格情報を再構成する手法を提案する．また，本研究では時間的空間的制約条件を考慮することで，骨格情報座標データの予測精度を向上させる．

第 3 章

ニューラルネットワーク

本章では，提案手法における使用されたニューラルネットワークの説明について記述する．まず，3.1 節では，敵対的生成ネットワークの概要，基本思想，loss 関数について説明する．次に，3.2 節では，長短期記憶ニューラルネットワークの構造，特徴，活性化関数について述べる．最後に，3.3 節では，パーセプトロンと多層パーセプトロンニューラルネットワークの概要，構成についてを記述する．

3.1 敵対的生成ネットワーク (GAN)

敵対的生成ネットワーク (Generative Adversarial Nets，GAN) は Goodfellow らによる提案されている 2 つのネットワークを競わせながら学習させるアーキテクチャである [40]．図 3.1 に GAN のアーキテクチャを示す．GAN は生成ネットワーク (Generator) と識別ネットワーク (Discriminator) に構成される．Generator は，生成データの特徴としてランダムノイズを入力することにより，所望のデータに近づけるように生成する．Discriminator は，

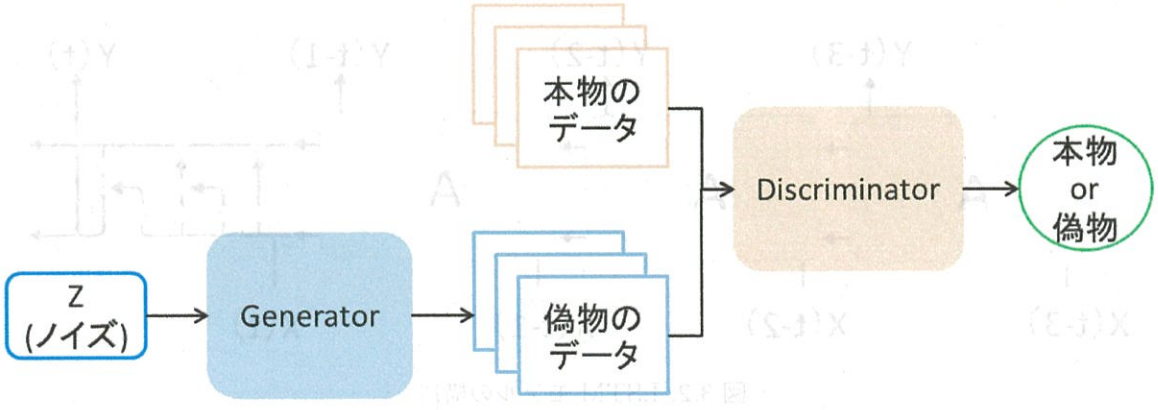


図 3.1. GAN のアーキテクチャ

Generator が生成した偽物のデータと本物のデータの真偽判定する．この 2 つのネットワークを交互に競合させることにより，Generator は本物のデータに近い偽物データを生成できるようになる．

Generator と Discriminator の競合関係は，数式 (3.1) の loss 関数を共有させることにより表現される [40]．右辺第 1 項は本物データを用いるケースであり，右辺第 2 項は Generator により生成されたデータである．まず，Generator を固定した上で Discriminator は数式 (3.1) を最大化する．本物データに対し，Discriminator は識別結果として 1 を出力させ，生成されたデータに対し，0 を出力させるように学習させる．次に，Discriminator を固定した上で Generator は数式 (3.1) を最小化する．Generator から生成した偽物の画像に，本物のラベルをつけて学習させる．

$$\underset{G}{max}\,\underset{D}{min}\,V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{3.1}$$

3.2　長短期記憶ニューラルネットワーク (LSTM)

GAN の生成ネットワークに用いる LSTM モデルについて説明する．長短期記憶ニューラルネットワー (Long Short-Term Memory，LSTM) は再帰型ニューラルネットワーク (Recurrent Neural Network，RNN) の一種であり，RNN の中間層のユニットを LSTM block と呼ばれるに置き換えることににによる実現されている [41]．LSTM の特長は，従来の RNN では学習できなかった長期依存 (long-term dependencies) を学習可能である．

図 3.2 に LSTM モデルの構造を示す．LSTM ユニット内部の情報の流れの 3 つのゲート (入力ゲート，出力ゲート，忘却ゲート) から構成される．LSTM のゲートの活性化関数にはロジスティック関数が使われることが多いが，学習速度を向上するために，図 3.3 に示すように，本研究では LeakyReLU 活性化関数が使われる．

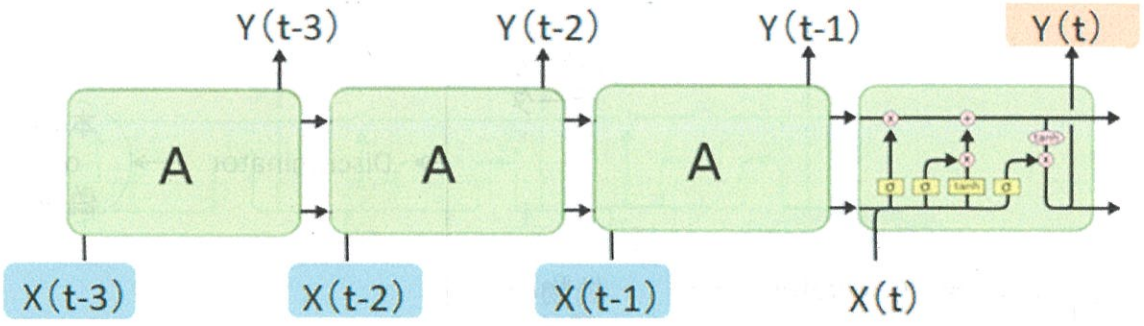


図 3.2. LSTM モデルの構造

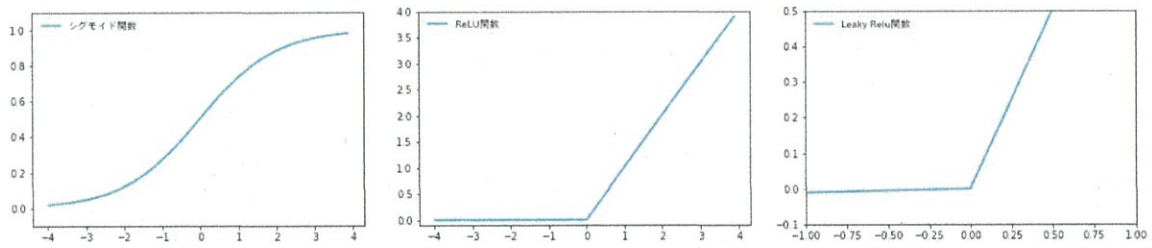


図 3.3. 活性化関数

3.3 多層パーセプトロンニューラルネットワーク (MLP)

GAN の識別ネットワークに用いる MLP モデルについて説明する．パーセプトロンは 2.2.1 項で簡単に説明したが，シンプルなアルゴリズムである (図 3.4)．パーセプトロンの入力は，単純特徴量と呼ばれる n 個の値による構成されるベクトル $[x_1, x_2, x_3, ..., x_n]$ である．出力は 1(yes) あるいは 0(no) である．単純パーセプトロンの活性化関数では，ステップ関数が使われ，数式 (3.2) のように定義されている．w は重みベクトルであり，wx は重みベクトルと入力ベクトルの内積であり，b はバイアスであり，1 か 0 しか出力できない．

$$f(x) = \begin{cases} 1 & wx + b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

多層パーセプトロン (Multi Layer Perceptron, MLP) は図 3.5 に示すように少なくとも 3 つのノードの層からなる階層型ニューラルネットワークである．1 つの入力層と，1 つ以上の中間層 (隠れ層) と 1 つの出力層から構成される．重み w とバイアス b の値は，ネットワークが出力と学習データ間の誤差を最小化するように最適化する．

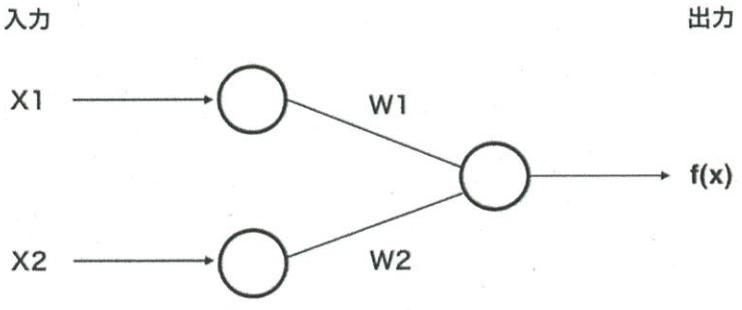


図 3.4. パーセプトロン

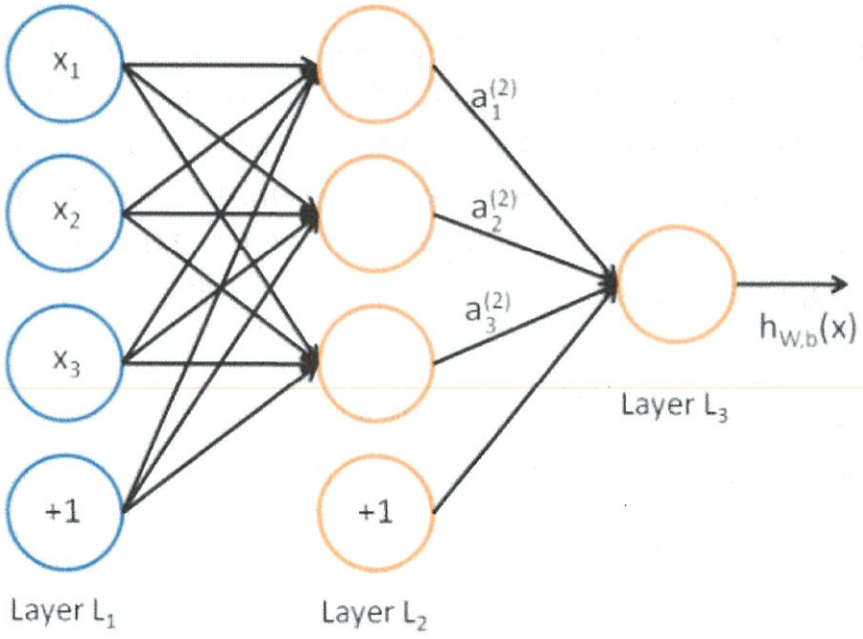


図 3.5. MLP モデルの構造

第 4 章

提案手法

4.1 遮蔽された人体骨格情報再構成の概要

本手法は，人体骨格情報の予測手法である．本研究が志向する遮蔽された人体骨格情報の再構成において，時間的空間的制約条件を考慮し，遮蔽された部位の骨格情報を復元する機能実現のために必要なコア技術である．

本手法では，単眼カメラが撮像でき，かつカメラに正対する歩く人，走る人，ジャンプする人と車椅子利用者を予測対象とする．遮蔽なしの動画をトレーニングデータとして，正対する時に人体骨格情報を最もよく取得できるという理由ばかりでない．例えば，インタラクティブシステムにおける姿勢認識の遅延の減少，遮蔽された姿勢を復元することのために，実際のアクションに先立って人物姿勢を推定する．また，ライブ中において，迷惑行動による記者，アイドル，重量挙げ選手などの撮影対象が遮蔽された場合に人物姿勢を復元する．さらに，エレベータや改札，階段等における高齢者の転倒予測，衝撃時における運転者と乗員の姿勢予測，工場における大きな荷物を台車で運ぶ時の危険予測等の適用も期待できる．

図 4.1 に提案手法のニューラルネットワーク構造を示す．本手法は，時間的空間的生成ネットワークと識別ネットワークによる構成されている．4.2 節で時間的空間的生成ネットワークと識別ネットワークの構造を説明する．深層ニューラルネットワークの利用のため，事前にモデルの訓練が必要であり，遮蔽なしの動画から取得した骨格情報座標データをトレーニングする．GAN のトレーニング方法が通常のニューラルネットワークと違うため，識別ネットワークを訓練してから生成ネットワークをトレーニングする．まず，撮影した動画から 1 フレームごとに骨格情報座標データを取得し，連続フレームの座標データと乱数行列を横方向に結合し，生成ネットワークによる連続フレームの座標データを生成する．次に，連続フレームの測定した座標データと生成した座標データを縦方向に結合し，識別ネットワークによる真偽の判定を学習させる．最後に，訓練した識別ネットワークの重みを固定し，生成ネットワークをトレニーングする．提案手法の手順を以下にまとめた．

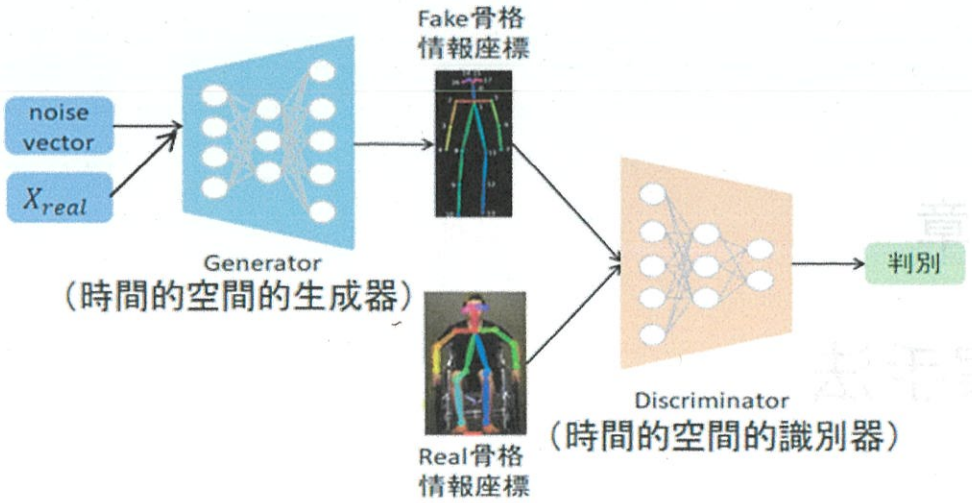


図 4.1. 提案手法の GAN の概要図

1. 撮影した動画から取得した連続フレームの骨格情報座標データを乱数行列と横方向に結合し，生成ネットワークにより，偽の座標データを生成
2. 測定した座標データと生成した座標データを縦方向に結合して真偽のラベル (1，0) を横方向に付け，識別ネットワークを訓練
3. 訓練した識別ネットワークの重みを固定し，生成ネットワークを用いて座標データを生成し，真のラベル (1) を横方向に付け，GAN を訓練
4. 訓練した GAN の生成ネットワークにより，遮蔽ありのフレームの骨格情報座標データを生成

4.2　時間的空間的生成ネットワークと識別ネットワーク

本節では，GAN における時間的空間的生成ネットワークと識別ネットワークの構造について述べる．

図 4.2 に生成ネットワークと識別ネットワークの構造を示す．GAN のパラメータとニューラルネットワーク構造を表 4.1 に示す．InputLayer は生成ネットワークの入力層であり，10 フレームの座標データと 1 フレームの乱数データを融合する．Model は生成ネットワークの出力層であり，5 フレームの座標データを予測する．Sequential は識別ネットワークである．Sequential の出力層は真又は偽の 1 個データのみを出力する．他の骨格情報推定技術を利用すれば，取得した座標データの数によって各層の Output Shape を設定する．

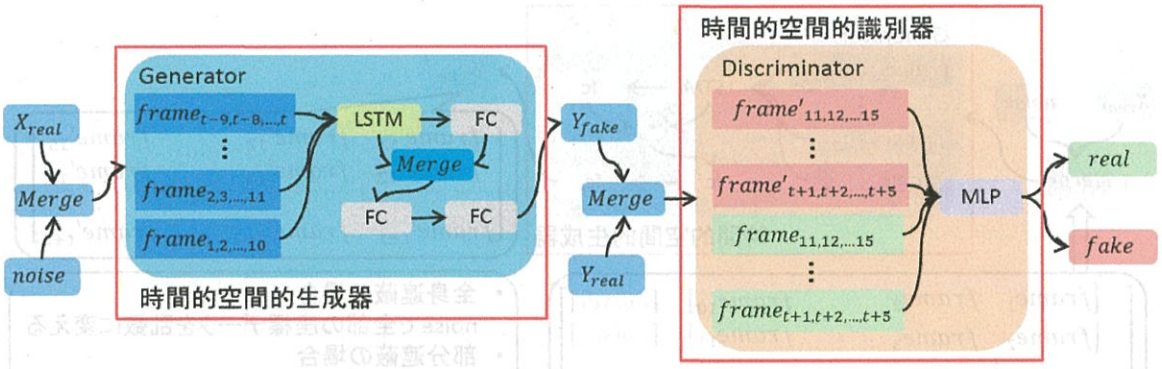


図 4.2. 生成ネットワークと識別ネットワークの構造

表 4.1. GAN のパラメータとニューラルネットワーク構造

Layer	Output Shape	Param
InputLayer	(None, 396)	0
Model	(None, 180)	846516
Sequential	(None, 1)	12697

4.2.1　生成ネットワーク

図 4.3 に示すように，生成ネットワークでは，連続フレームの座標データ行列と乱数行列を横方向に結合し，LSTM モデルに入力し，FC 層に入力する．LSTM の出力行列と FC 層の出力行列のアダマール積を計算し (数式 4.1 による計算する)，2 層の FC 層に入力し，連続フレームの座標データを生成する．

全身遮蔽の場合，乱数行列に全ての座標データを乱数に変える．部分遮蔽の場合，乱数行列に遮蔽された部位のみの座標データを乱数に変え，他の遮蔽されていない部位の座標データは測定値を利用する．ニューラルネットワークのトレーニングにおいて，10 フレームの座標データと 1 フレームの乱数データを横方向に結合し，生成ネットワークに入力する．そして，LSTM の出力行列と FC 層の出力行列のアダマール積を計算し，2 層の FC 層に入力する．最後に，5 フレームの座標データを予測する．生成ネットワークのパラメータとニューラルネットワーク構造を表 4.2 に示す．

$$\boldsymbol{A} \circ \boldsymbol{B} = (\boldsymbol{a_{ij}}) \cdot (\boldsymbol{b_{ij}})_{1 \le i \le m, 1 \le j \le n} \tag{4.1}$$

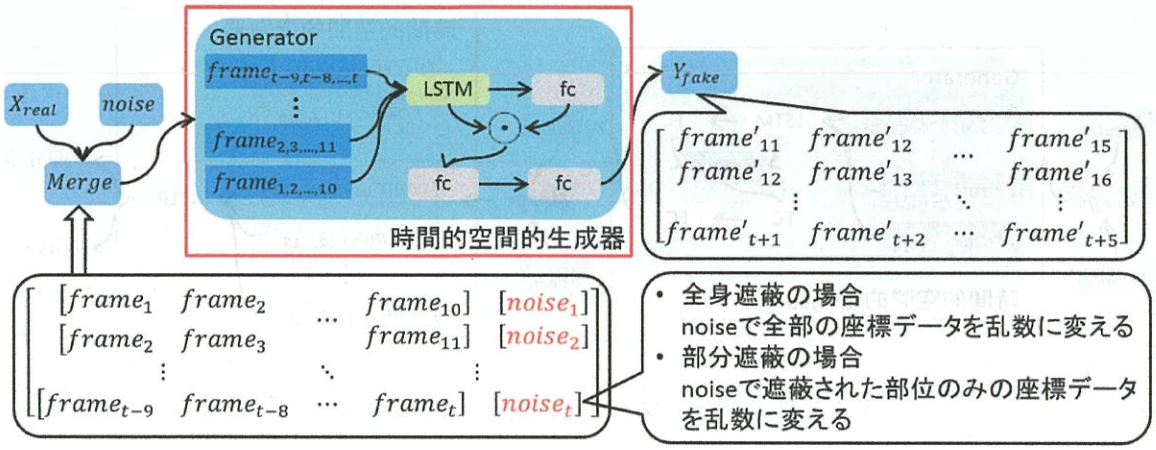


図 4.3. 生成ネットワークにおける骨格情報座標データの処理図

表 4.2. 生成ネットワークのパラメータとニューラルネットワーク構造

Layer	Output Shape	Param
InputLayer	(None, 1, 396)	0
LSTM	(None, 256)	668672
Dense	(None, 256)	65792
Multiply	(None, 256)	0
Dense	(None, 256)	65792
Dense	(None, 180)	46260

骨格情報座標データと乱数行列を融合する前に，精度を上げるため，前処理が必要であり，データ正規化を行う．特徴量として，動画からの骨格情報座標データの値の範囲を [0,1] の範囲におさめる．乱数行列では，[0,1] の範囲に一様乱数を使い，ループで異なる乱数を使用する．LSTM 層において LeakyReLU 活性化関数を使い，FC 層において sigmoid 活性化関数を利用する．

生成ネットワークにおいて，LSTM 層の出力行列と FC 層の出力行列のアダマール積を計算する理由は，Attention モデルの計算方法の一種を使用するためである．Attention モデルはベクトルの加重平均を求め，出力要素と強く相関する入力要素を抽出する仕組みである．画像認識，自然言語処理分野の研究において多く使われており，高精度に分類できることが報告されている [42, 43, 44, 45]．また，通常の Attention モデルにおいて，softmax 活性化関数が使われているが，He らが提案されている Mask R-CNN において，sigmoid 活性化関数が使われ，更に高精度に分類できることが報告されている [46]．本研究では，カテゴリーの分類を考慮しないため，sigmoid 活性化関数を使用する．

4.2.2 識別ネットワーク

図 4.4 に示すように，識別ネットワークでは，生成された連続フレームの座標データと現実的な連続フレームの座標データを縦方向に結合して真偽のラベルを付け，MLP モデルに入力し，データ真偽の判定を学習させる．

MLP に入力する前に，5 フレームの生成された座標データと現実的な 5 フレームの座標データを縦方向に結合し，生成データの横方向に 0(偽) ラベルを付け，測定データの横方向に 1(真) ラベルを付け，ラベルにランダムノイズを追加する．中間層において LeakyReLU 活性化関数を使用する．出力層は sigmoid 活性化関数を使い，真偽の 2 クラス分類を行う．識別ネットワークのパラメータとニューラルネットワーク構造を表 4.3 に示す．

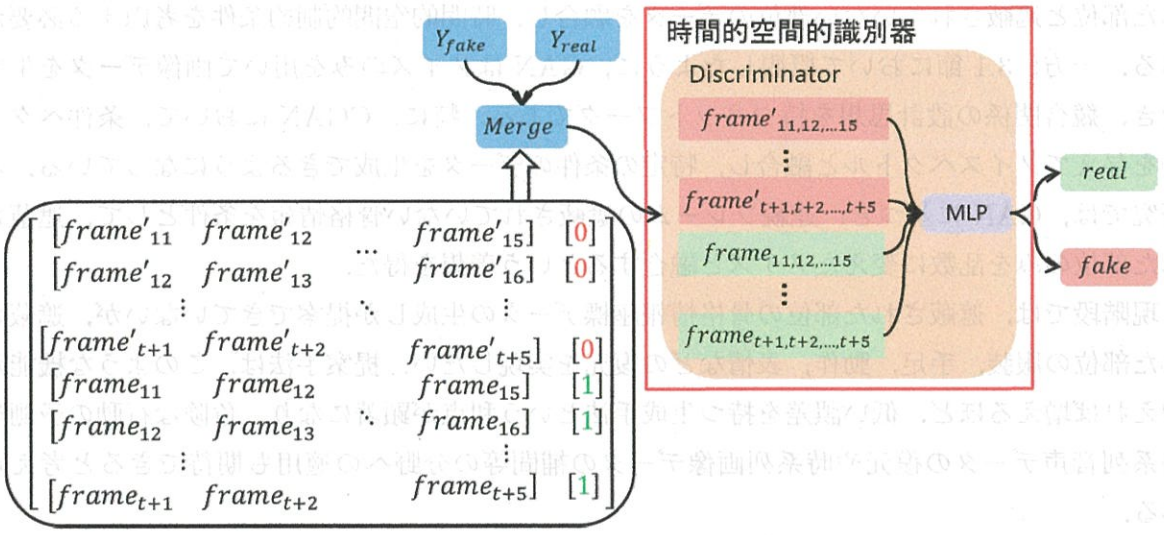


図 4.4. 識別ネットワークにおける骨格情報座標データの処理図

表 4.3. 識別ネットワークのパラメータとニューラルネットワーク構造

Layer	Output Shape	Param
Dense	(None, 64)	11584
Dense	(None, 16)	1040
Dense	(None, 4)	68
Dense	(None, 1)	5

4.3　提案手法の特徴

提案手法の特徴は，時間的空間的制約条件を考慮した骨格情報座標データを生成することである．通常，人物の姿勢推定において，遮蔽が発生した場合，時間的制約条件のみを考慮し，遮蔽されていない部位の座標データを活用せず，骨格情報座標データを予測する．本研究では，遮蔽されていない部位の座標データを併用し，予測精度を向上させることを期待できる．また，従来に LSTM と Attention モデルは動作認識，自然言語分類等の分類技術に多く利用されている．本研究では，骨格情報の予測に対し，GAN，LSTM，Attention モデルを利用し，モデルの有効性を検証する．

時間的制約条件だけではなく，遮蔽されていない部位の骨格情報を併用するために，遮蔽された部位と遮蔽されていない部位のデータを融合し，時間的空間的制約条件を考慮する必要がある．一方，3.1 節において概観したように，GAN はノイズのみを用いて画像データを生成でき，競合関係の設計思想を持つネットワークである．特に，CGAN において，条件ベクトルを与えてノイズベクトルと融合し，特定の条件のデータを生成できるようになっている．本研究では，GAN に基づき，連続フレームの遮蔽されていない骨格情報を条件として，遮蔽された部位のみを乱数に変えたノイズと融合するという着想を得た．

現階段では，遮蔽された部位の骨格情報座標データの生成しか提案できていないが，遮蔽された部位の服装，手足，動作，表情などの復元を実現したい．提案手法は，このような機能が増えれば増えるほど，低い誤差を持つ生成手法という利点が顕著になり，危険な行動の予測や時系列音声データの復元や時系列画像データの補間等の分野への適用も期待できると考えられる．

第 5 章

評価と考察

5.1 実験概要

本節では，実験環境，データセット，モデルのパラメータ，ニューラルネットワーク構造について述べる．

本研究の有効性を検証するために，本研究で作成したモデル，GRU-GAN，CNN-LSTM，LSTM，MLP の五つモデルの予測誤差を比較し，評価実験を行った．テスト用のサンプルデータとして，公開データセット MoCap と独自に撮影した車椅子利用者動画を利用し，骨格情報推定技術の OpenPose に基づき，取得した骨格情報の測定データと生成した予測データの予測誤差を比較した．開発環境のハードウェアとソフトウェアについては表 5.1 に示す．

表 5.1. 実験環境

OS	Windows 8(IDE),Window 10(Openpose)
IDE	PyCharm 2019.3
Programming	Python 3.6
Openpose	Version 1.3
GPU	GeForce GTX 1080(8G)
Library	Keras 2.2,Matplotlib 2.2,Numpy 1.14,Opencv-python 4.1, Pandas 0.23,Pip 19.2,Scikit-learn 0.19,Scipy 1.1,Tensorflow 1.9

5.1.1　MoCap データセット

MoCap データセットはカーネギーメロン大学モーションキャプチャデータベースであり，6 つのカテゴリと 23 のサブカテゴリに 2605 件の撮影した動画 (352×240pixels，30fps) がある．

本研究では，MoCap データセットにおけるカメラに正対する歩く人，走る人とジャンプする人の 3 種類動作の 24 本動画を使用し，図 5.1〜5.3 に示すような画像に対して測定座標データと予測座標データの誤差を比較し，シミュレーションを行った．表 5.2 に使用された動画のカテゴリー，サブカテゴリ，トレーニング又はテスト，ID 番号とフレーム数を示す．全てのトレーニング用のフレーム数は 1151，テスト用のフレーム数は 966 である．

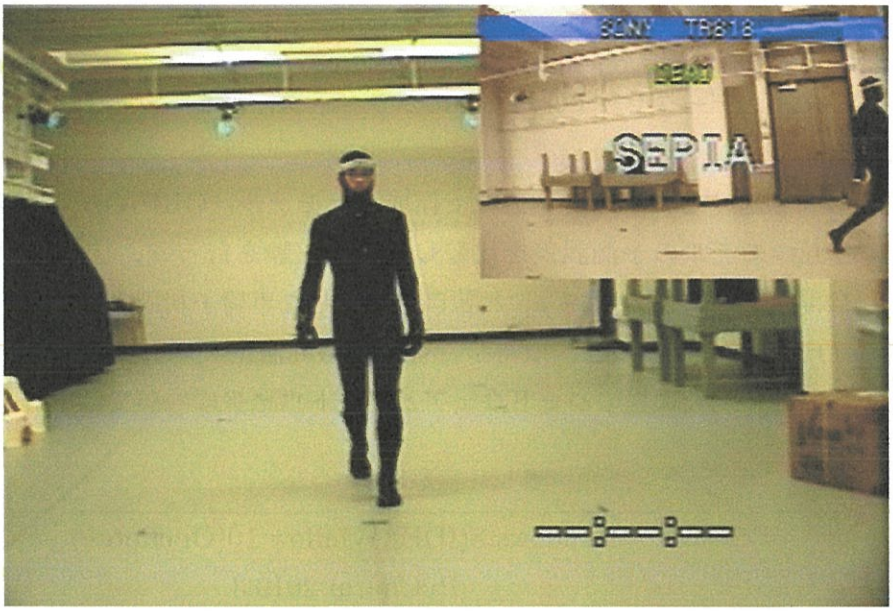


図 5.1. MoCap データセットの歩く人の画像例

図 5.2. MoCap データセットの走る人の画像例

図 5.3. MoCap データセットのジャンプする人の画像例

表 5.2. 使用された MoCap データセット動画の詳細情報

Category	Subcategory	Train or Test	ID	使用するフレーム数
Locomotion	Walk	Train	16-15	199
			16-16	204
			16-31	107
			16-32	142
		Test	16-21	133
			16-22	139
			16-47	158
			16-58	144
	Run	Train	16-8	78
			16-35	78
			16-36	77
			16-45	26
		Test	16-46	39
			16-55	30
			16-56	37
			16-57	46
	Jump	Train	13-39	60
			13-41	60
			16-1	60
			16-3	60
		Test	13-40	60
			13-42	60
			16-2	60
			16-4	60

5.1.2 車椅子利用者データ

表 5.3 に車椅子利用者データの作成条件を示し，表 5.4 に撮影条件を示す．車椅子は日本工業規格 JIS T 9201 に定める規格サイズに準じたものを使用した (図 5.1)．表 5.3 に示すように，撮影パターンは車椅子利用時では 1 パターン (カメラに正対する直進走行) を撮影した．

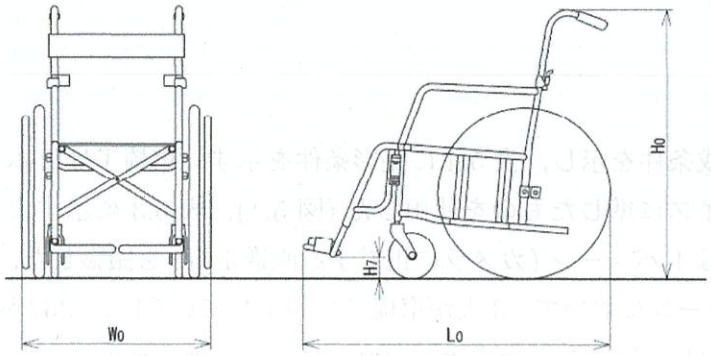
本研究では，表 5.3 に示す撮影パターンにおいて，3 人が車椅子を利用している時，1332 枚，243 枚，273 枚のフレーム動画を撮影し，図 5.5 に示すような画像に対して測定座標データと予測座標データの誤差を比較し，シミュレーションを行った．

表 5.3. 車椅子利用者データの作成条件

	協力者 A	協力者 B	協力者 C
性別	男	男	男
身長 (cm)	177	174	173
体格	痩せ型	標準	中肉
撮影フレーム数	1332	243	273
Train or Test	Train	Test	Test

表 5.4. 撮影条件

画角	73 deg
解像度	1,920 × 1,080 pixels
フレームレート	30 fps
設置高さ	2.4 m
設置俯角	7 deg
撮影場所	本学日野キャンパス 2 号棟 8F 廊下 (幅 2.8m, 長さ 30m)



部位	寸法値 b)
全長（L_0）	1200 以下
全幅（W_0）	700 以下
フットサポート高（H_7）	50 以上
全高（H_0） a)	1200 以下

a）ヘッドサポートを外した時
b）リクライニング機構及び／又はティルト機構を装備する車椅子は、標準状態の寸法とする。

図 5.4. 車椅子の規格

図 5.5. 車椅子利用者データの画像例

5.1.3　各比較モデルのパラメータと構造

提案手法，GRU-GAN，CNN-LSTM，LSTM，MLP の 5 つのモデルを比較するために，データ正規化手法，ユニット数，dropout(0.2)，活性化関数 (LeakyReLU，sigmoid)，loss 関数 (mae，binary-crossentropy)，optimizer(adam，RMSprop) を可能な限り同様に設定した．詳細なコード断片については，付録に記述する．

提案手法の GAN のパラメータと構造は 4.2 節に記述があるが，GRU-GAN，CNN-LSTM，LSTM，MLP の 4 つモデルのパラメータと構造は表 5.5〜5.10 に示す．なお，学習の繰り返し数である epoch は図 5.6〜41 に示す横軸 epoch と縦軸 loss の推移グラフにより，loss 曲線が安定して収束する epoch の範囲を観測できる．各モデルの epoch 数の設定は表 5.11 に示す．

表 5.5. GRU-GAN の生成ネットワークのパラメータと構造

Layer	Output Shape	Param
GRU	(None, 256)	501504
Dense	(None, 256)	65792
Dense	(None, 180)	46260

表 5.6. GRU-GAN の識別ネットワークのパラメータと構造

Layer	Output Shape	Param
Dense	(None, 64)	11584
Dense	(None, 16)	1040
Dense	(None, 4)	68
Dense	(None, 1)	5

表 5.7. GRU-GAN の GAN のパラメータと構造

Layer	Output Shape	Param
InputLayer	(None, 1, 396)	0
Sequential	(None, 180)	613556
Sequential	(None, 1)	12697

表 5.8. CNN-LSTM のパラメータと構造

Layer	Output Shape	Param
InputLayer	(None, 1, 360)	0
Conv1D	(None, 1, 256)	92416
MaxPooling1D	(None, 1, 256)	0
Conv1D	(None, 1, 128)	32896
MaxPooling1D	(None, 1, 128)	0
Conv1D	(None, 1, 64)	8256
LSTM	(None, 256)	328704
Dense	(None, 256)	65792
Multiply	(None, 256)	0
Dense	(None, 36)	9252

表 5.9. LSTM のパラメータと構造

Layer	Output Shape	Param
LSTM	(None, 256)	631808
Dense	(None, 256)	65792
Dense	(None, 36)	9252

表 5.10. MLP のパラメータと構造

Layer	Output Shape	Param
Dense	(None, 64)	23104
Dense	(None, 16)	1040
Dense	(None, 4)	68
Dense	(None, 36)	180

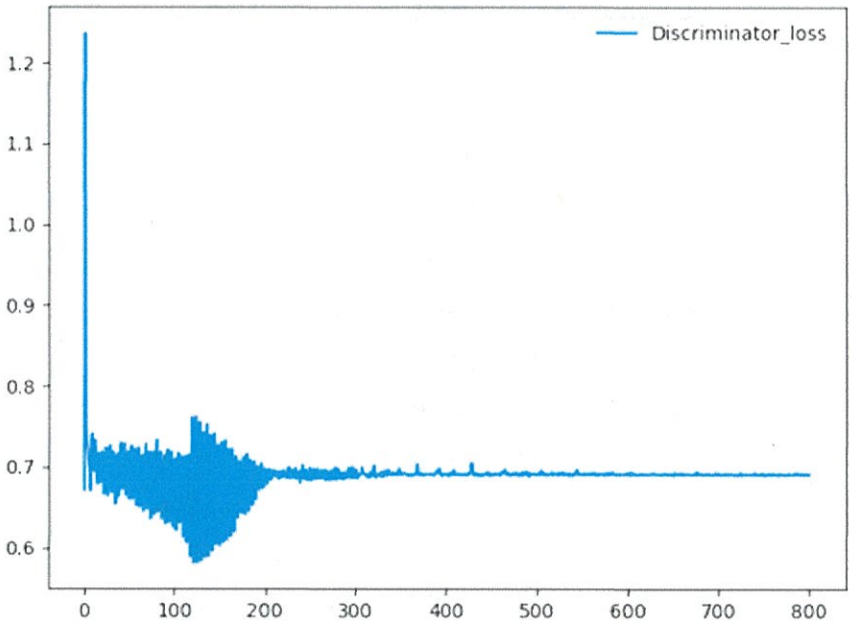


図 5.6. 歩く人データにする提案手法の GAN の識別ネットワークの loss

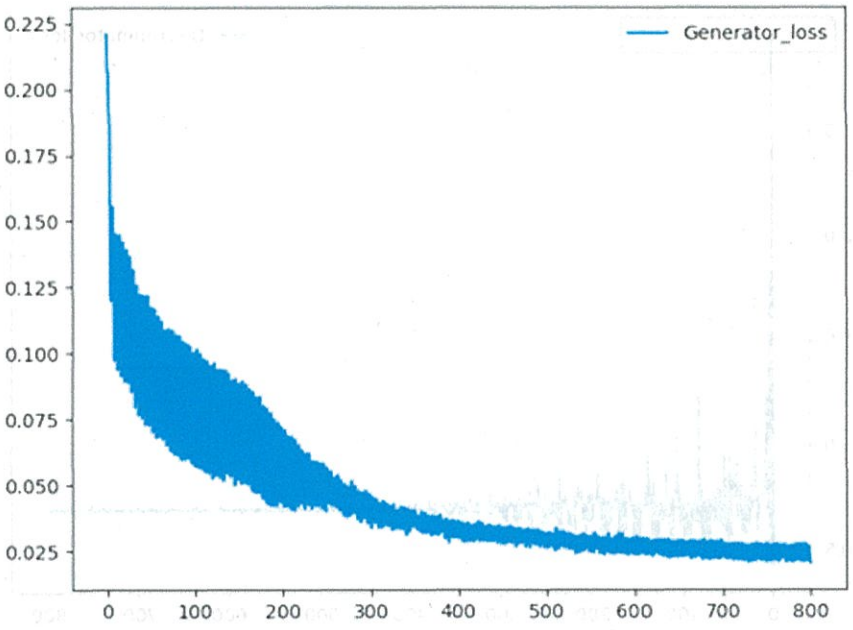


図 5.7. 歩く人データにする提案手法の GAN の生成ネットワークの loss

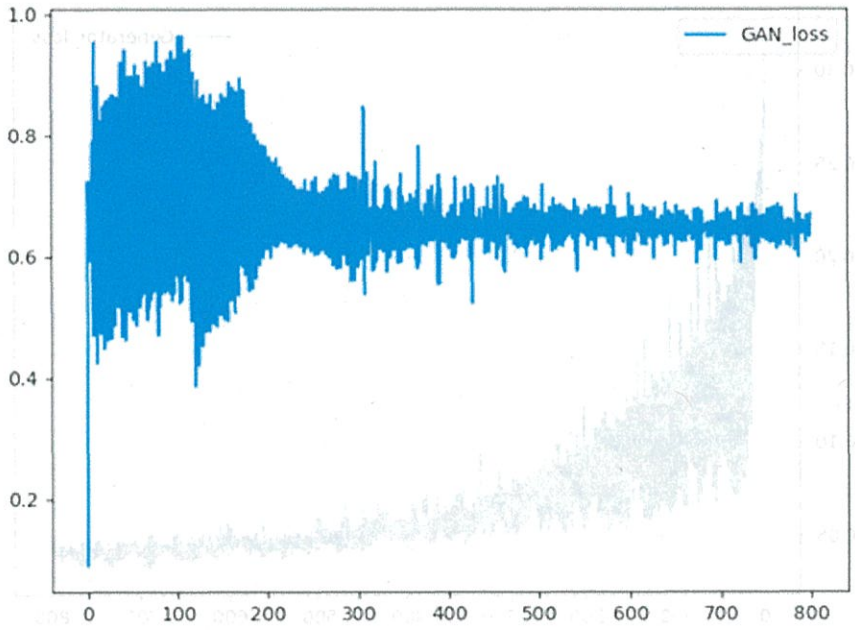


図 5.8. 歩く人データにする提案手法の GAN の loss

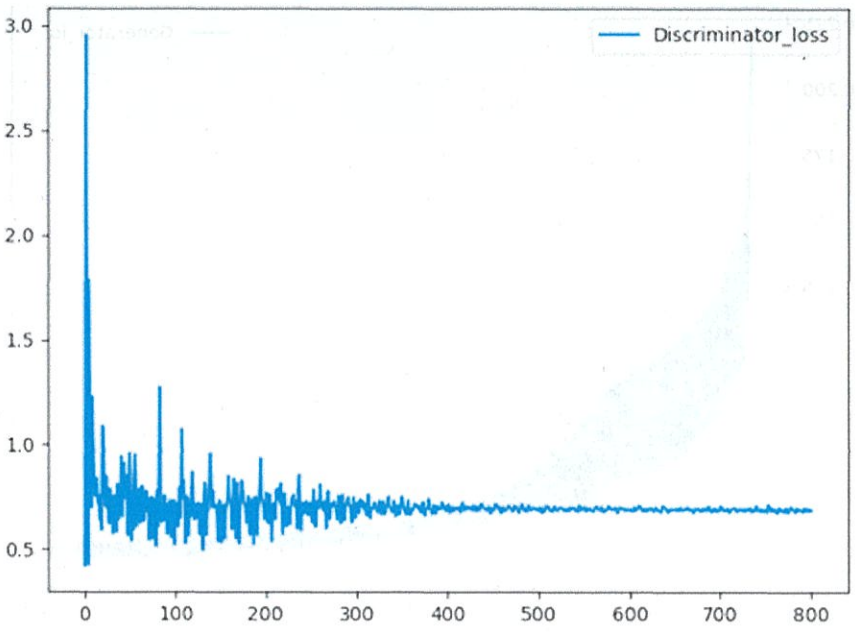


図 5.9. 走る人データにする提案手法の GAN の識別ネットワークの loss

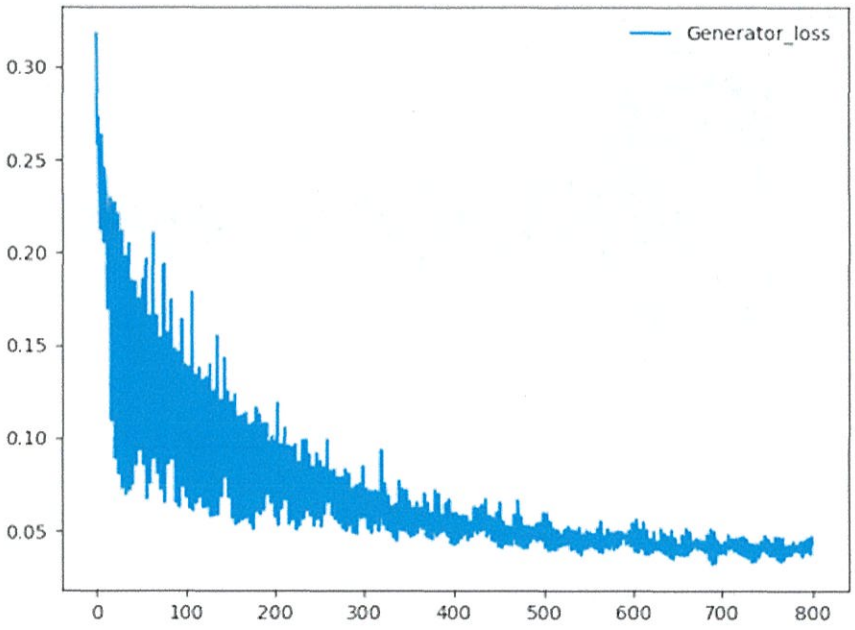


図 5.10. 走る人データにする提案手法の GAN の生成ネットワークの loss

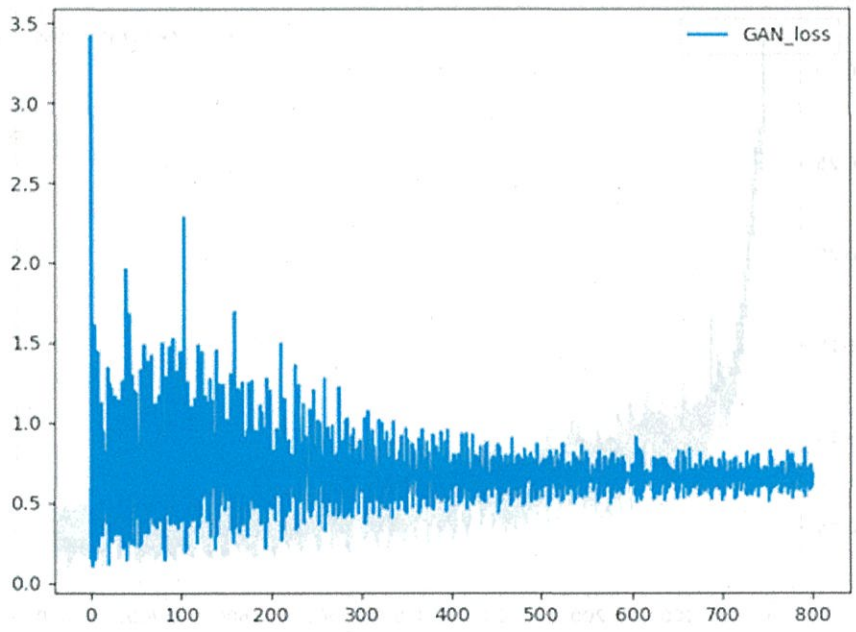


図 5.11. 走る人データにする提案手法の GAN の loss

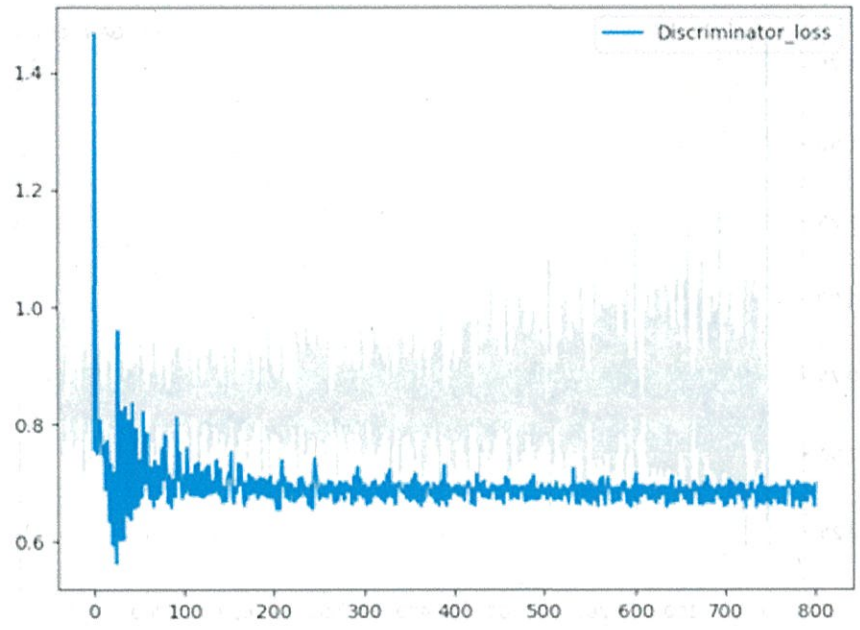


図 5.12. ジャンプする人データにする提案手法の GAN の識別ネットワークの loss

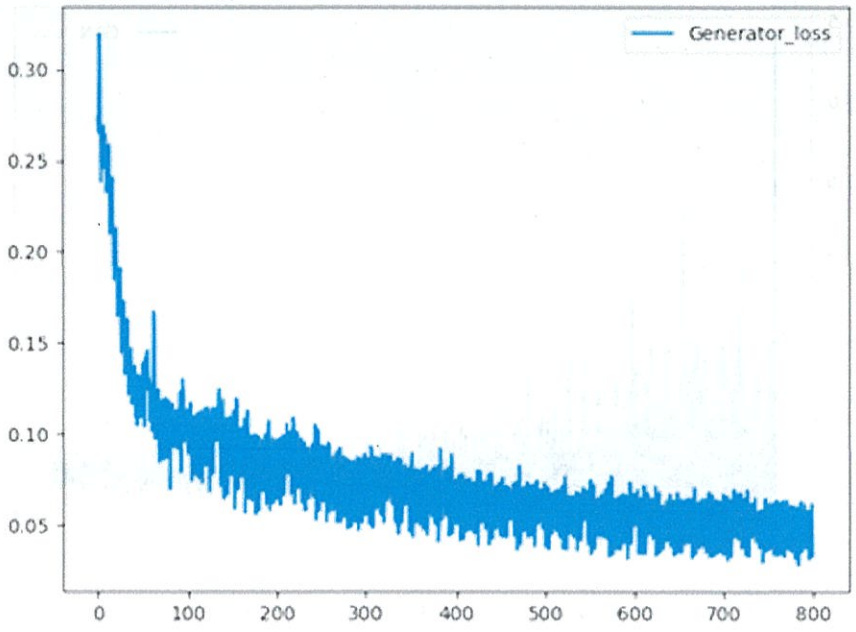


図 5.13. ジャンプする人データにする提案手法の GAN の生成ネットワークの loss

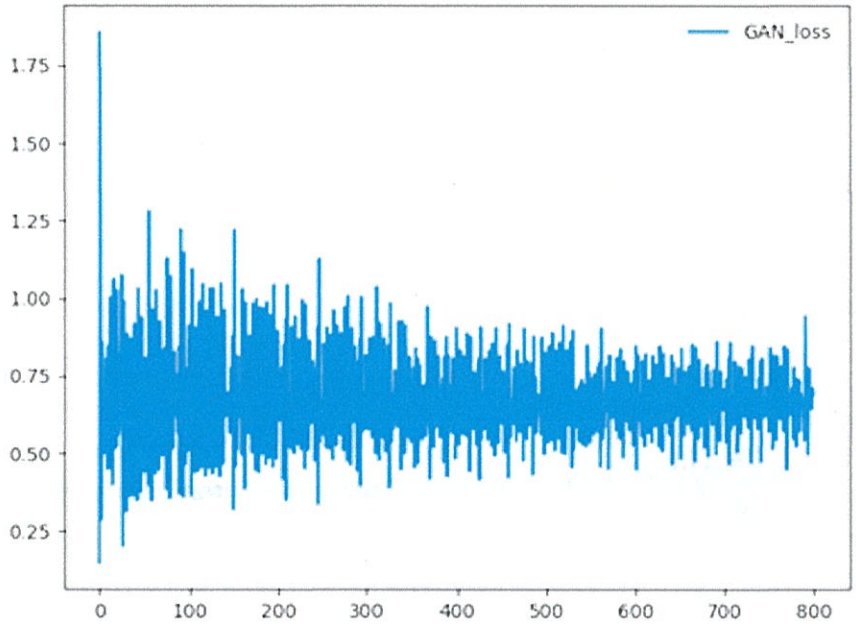


図 5.14. ジャンプする人データにする提案手法の GAN の loss

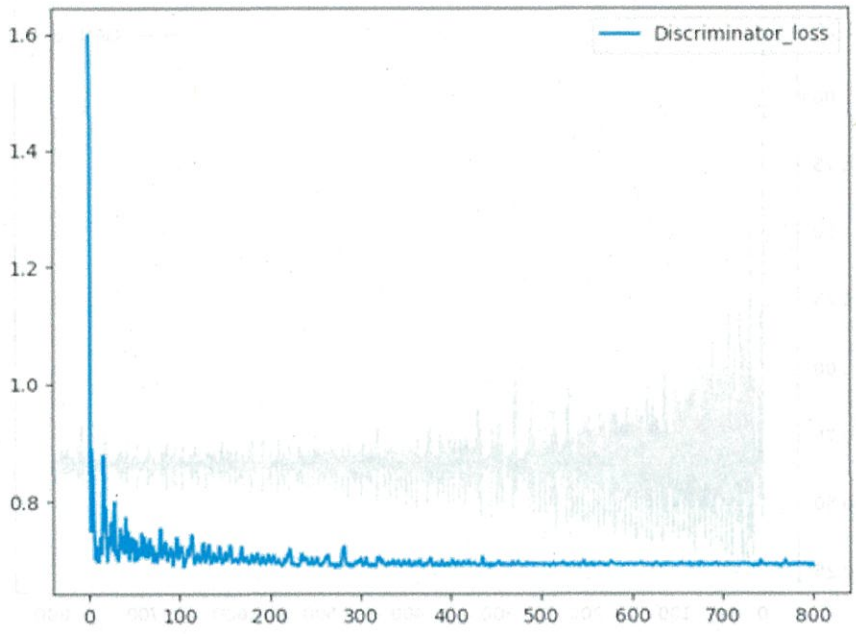


図 5.15. 車椅子利用者データにする提案手法の GAN の識別ネットワークの loss

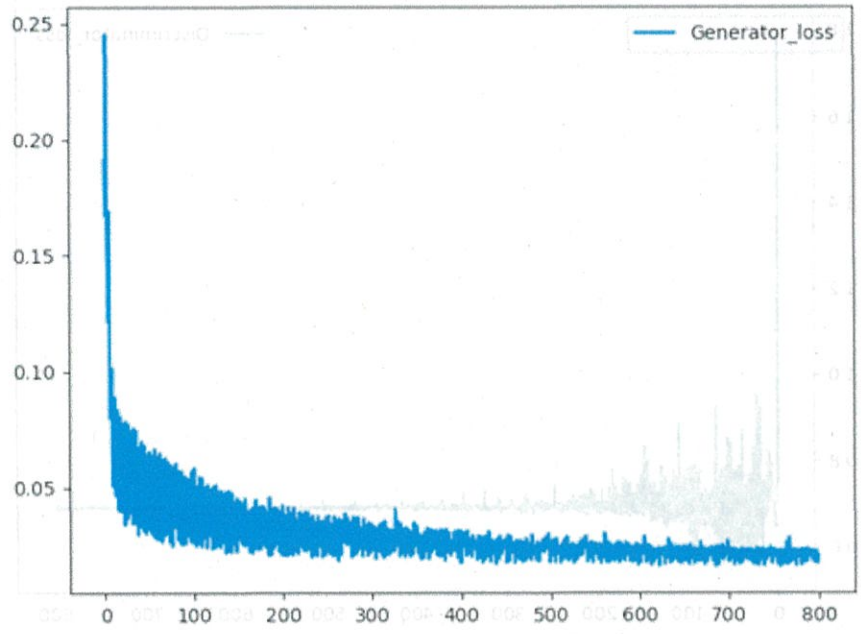


図 5.16. 車椅子利用者データにする提案手法の GAN の生成ネットワークの loss

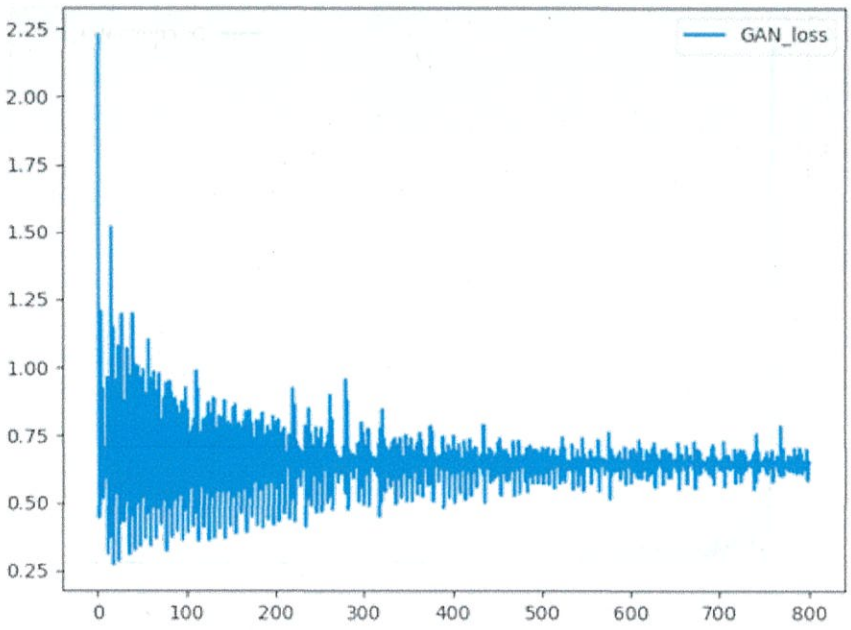


図 5.17. 車椅子利用者データにする提案手法の GAN の loss

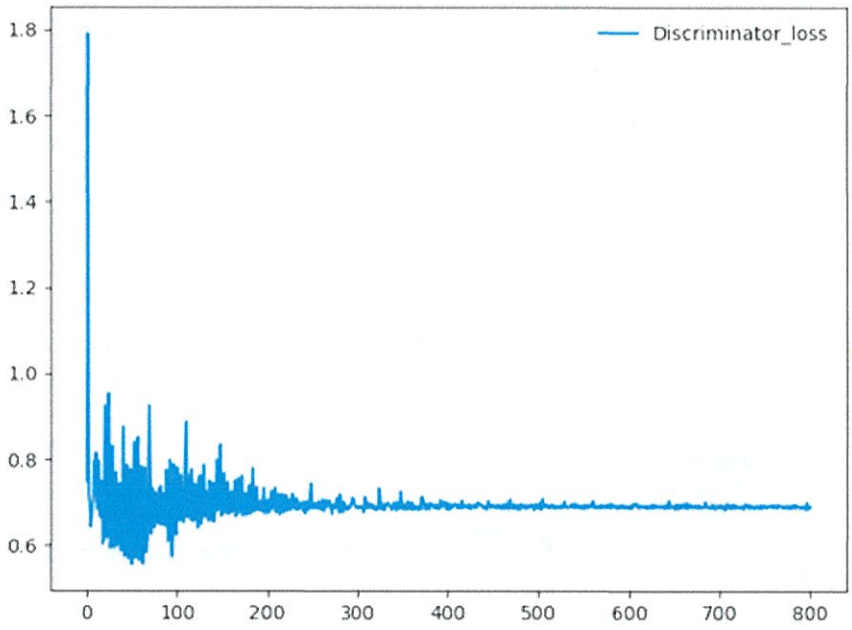


図 5.18. 歩く人データにする GRU-GAN の識別ネットワークの loss

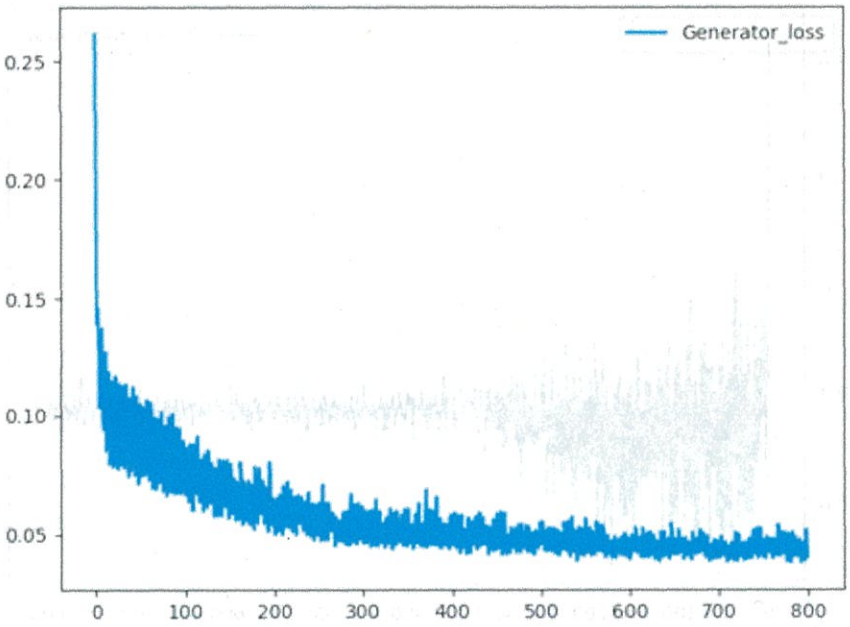


図 5.19. 歩く人データにする GRU-GAN の生成ネットワークの loss

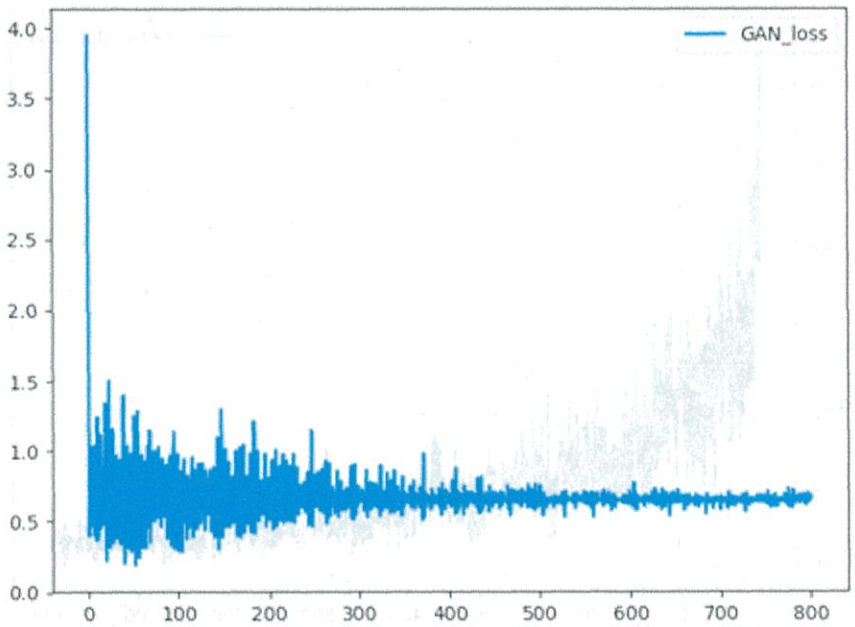


図 5.20. 歩く人データにする GRU-GAN の loss

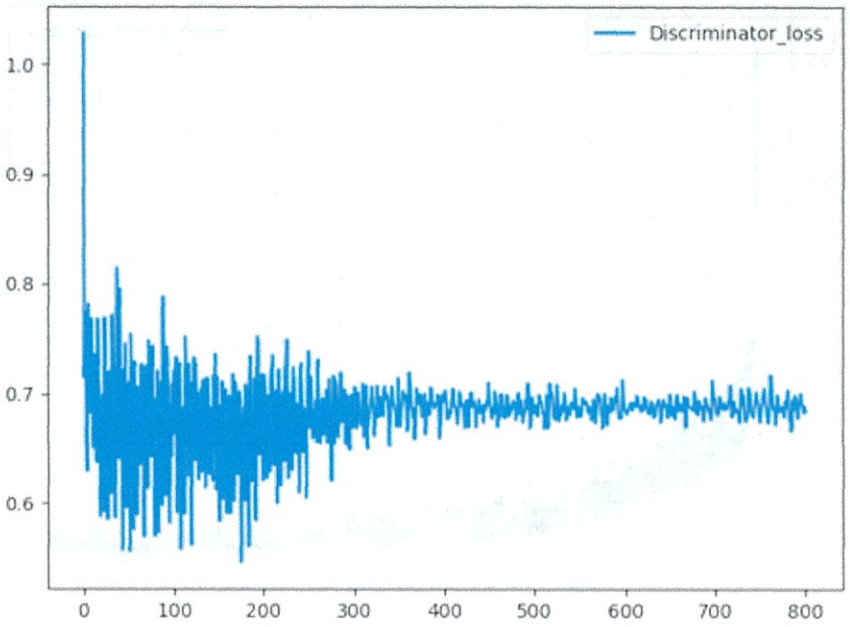


図 5.21. 走る人データにする GRU-GAN の識別ネットワークの loss

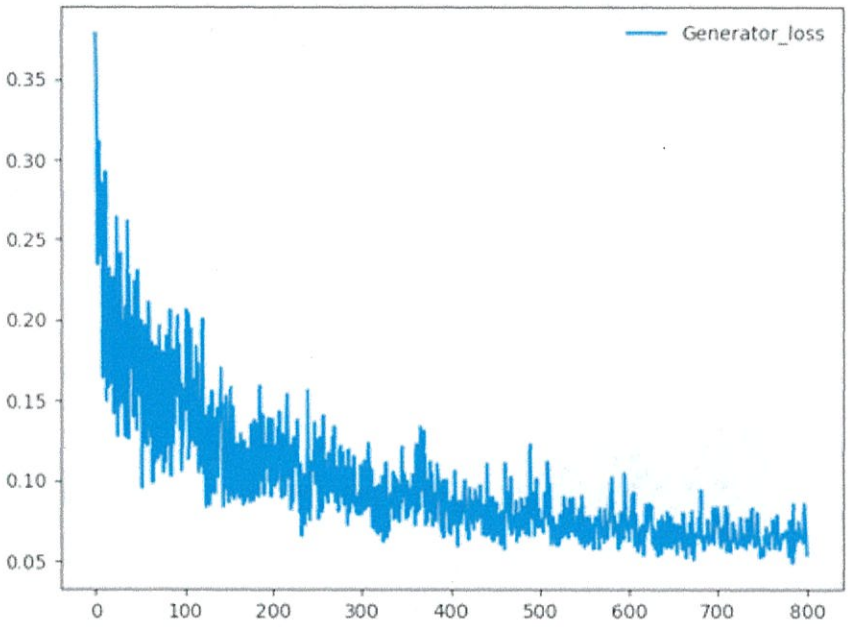


図 5.22. 走る人データにする GRU-GAN の生成ネットワークの loss

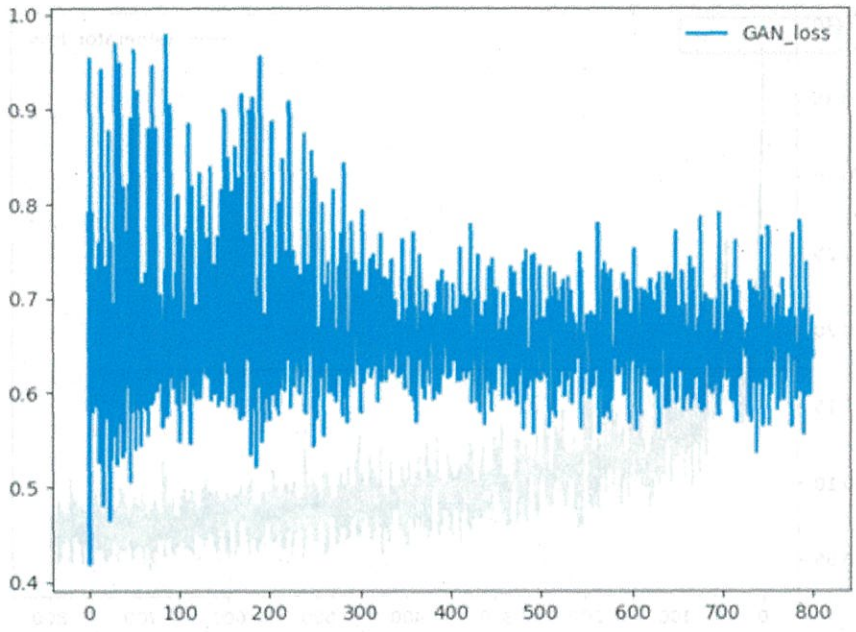


図 5.23. 走る人データにする GRU-GAN の loss

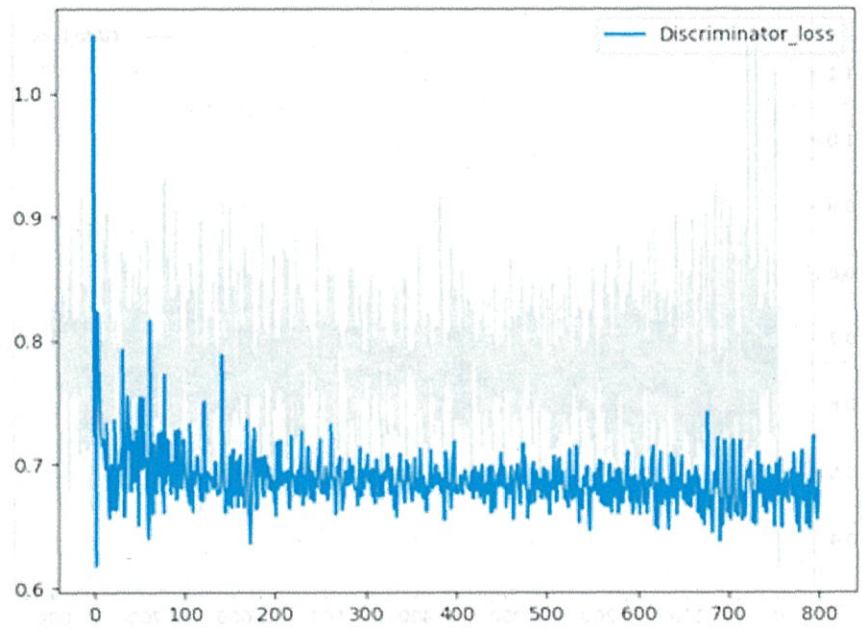


図 5.24. ジャンプする人データにする GRU-GAN の識別ネットワークの loss

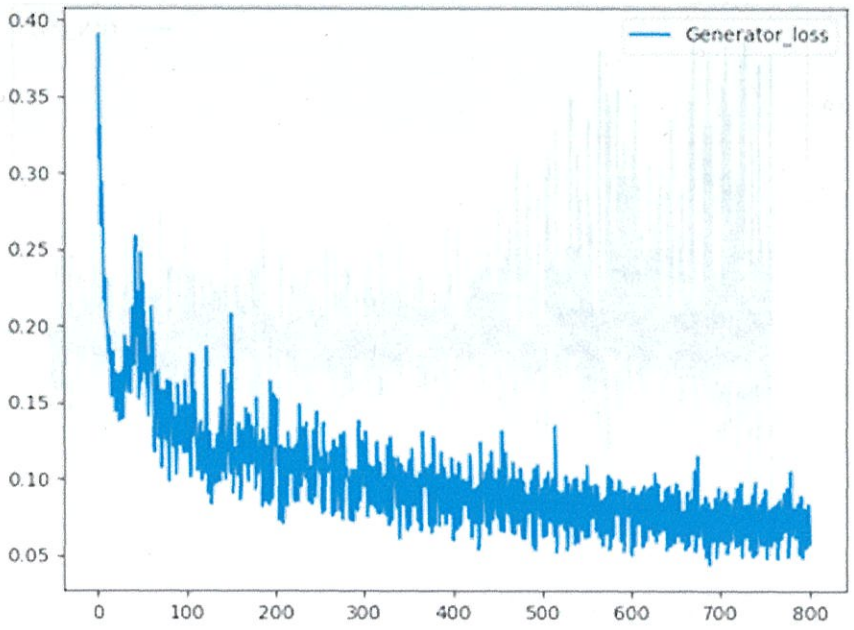


図 5.25. ジャンプする人データにする GRU-GAN の生成ネットワークの loss

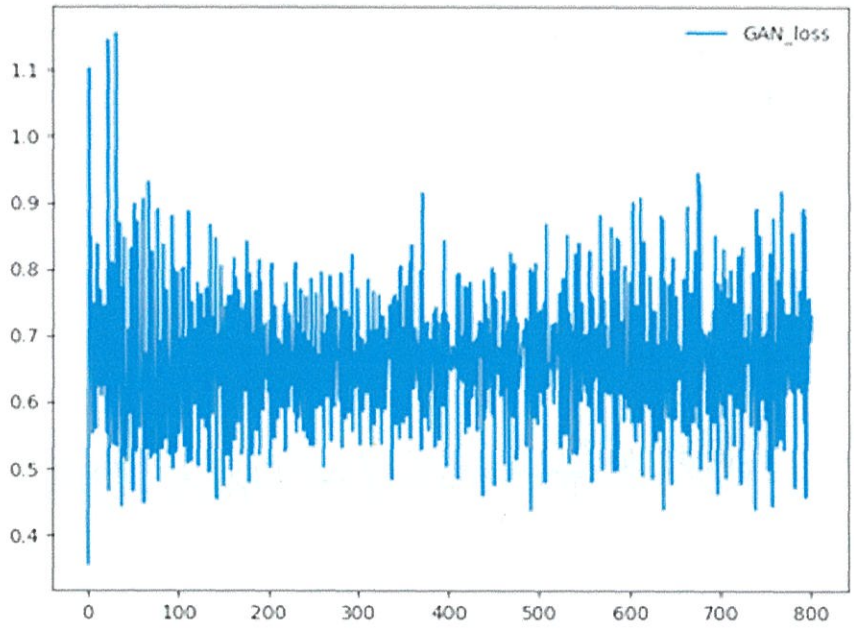


図 5.26. ジャンプする人データにする GRU-GAN の loss

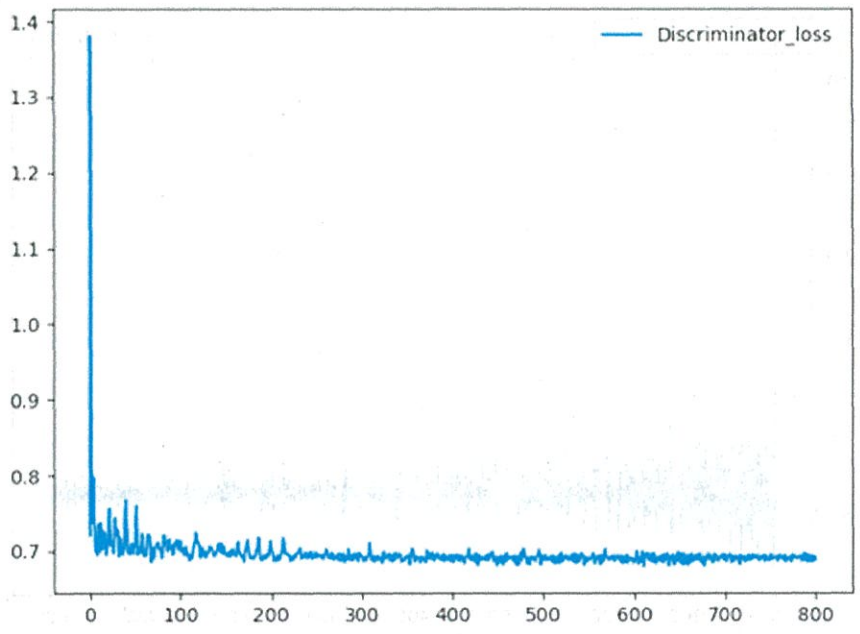


図 5.27. 車椅子利用者データにする GRU-GAN の識別ネットワークの loss

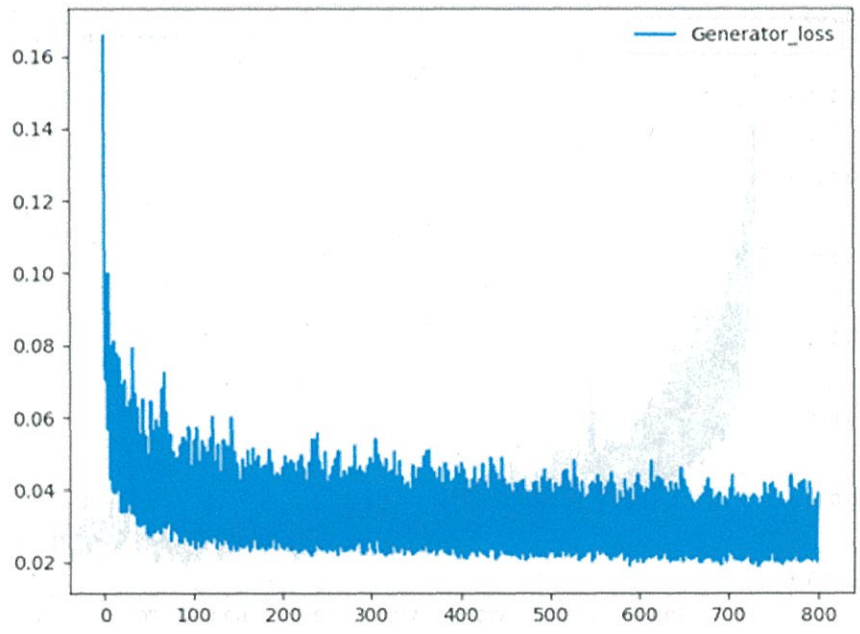


図 5.28. 車椅子利用者データにする GRU-GAN の生成ネットワークの loss

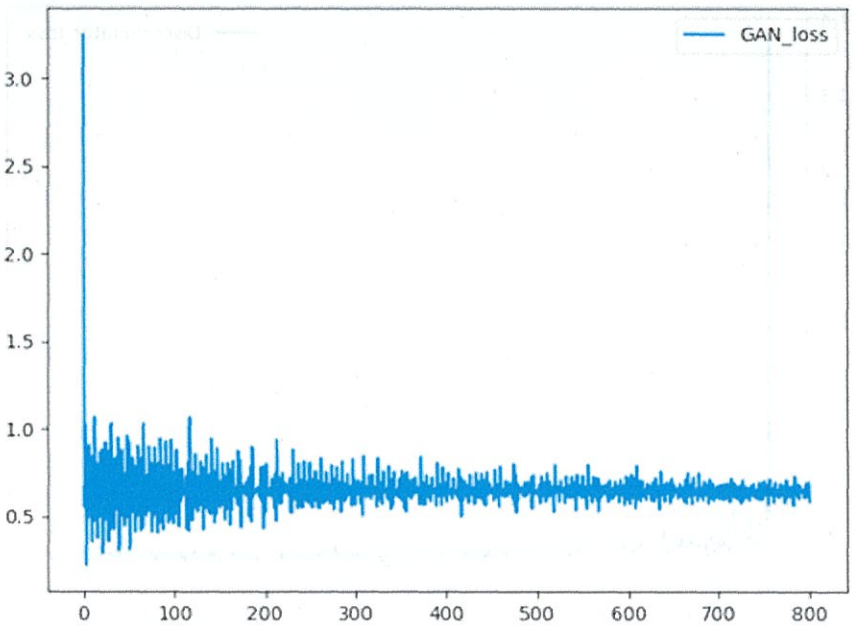


図 5.29. 車椅子利用者データにする GRU-GAN の loss

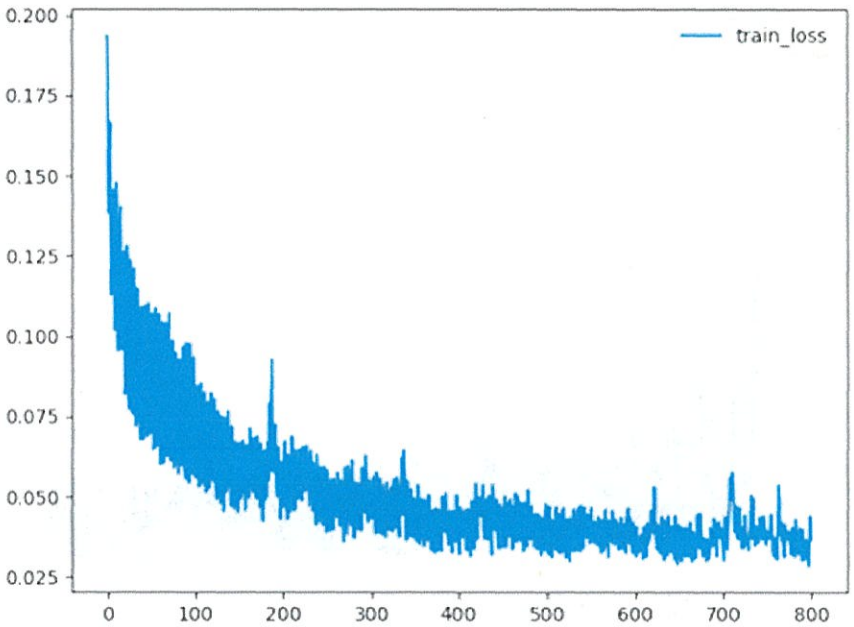


図 5.30. 歩く人データにする CNN-LSTM の loss

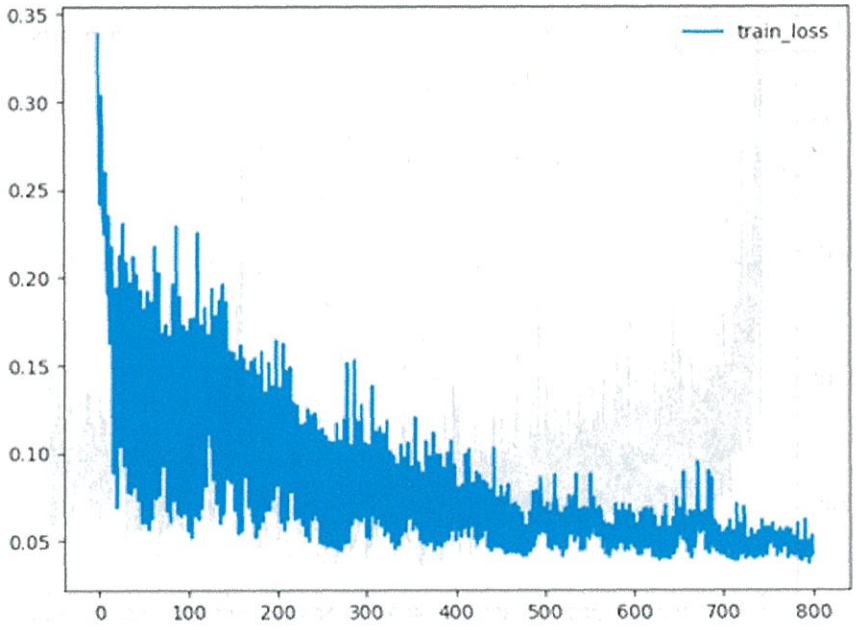


図 5.31. 走る人データにする CNN-LSTM の loss

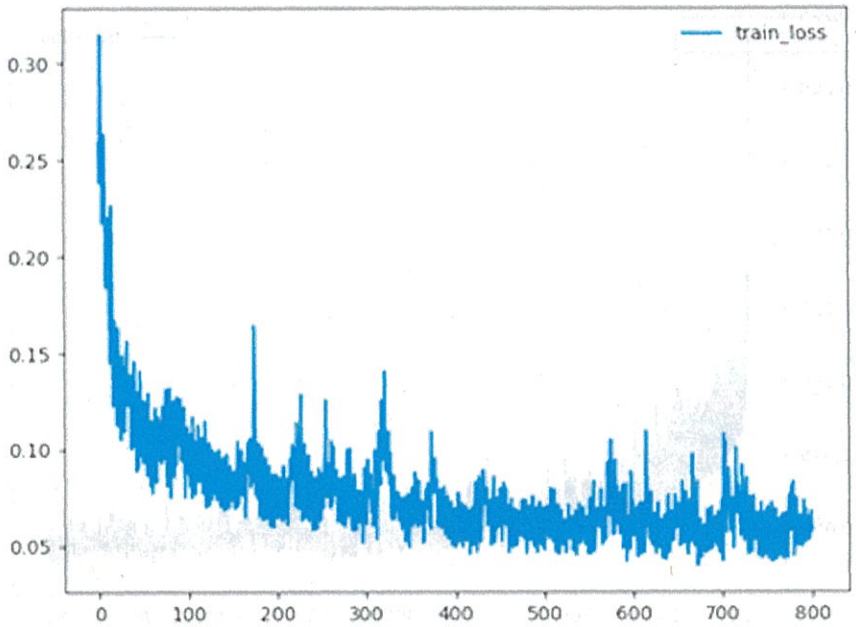


図 5.32. ジャンプする人データにする CNN-LSTM の loss

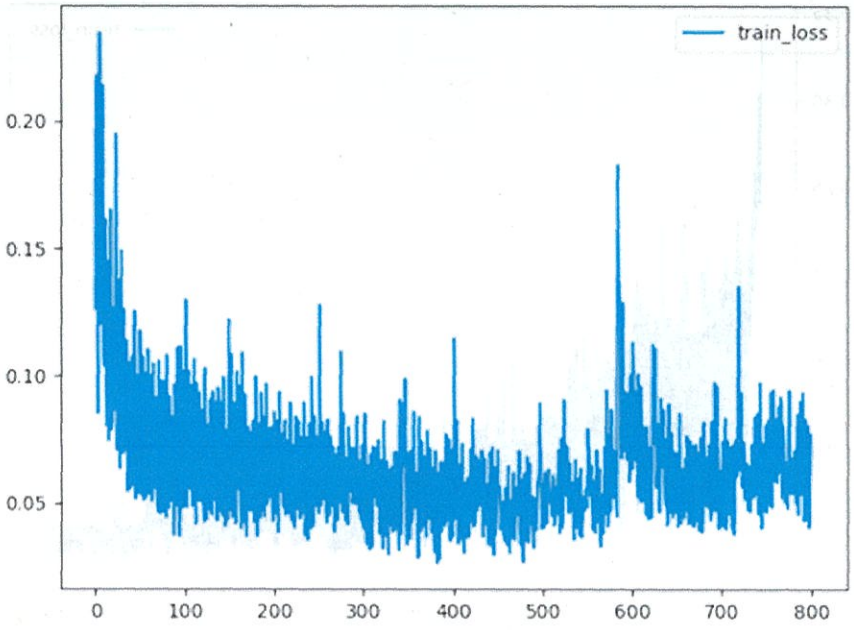


図 5.33. 車椅子利用者データにする CNN-LSTM の loss

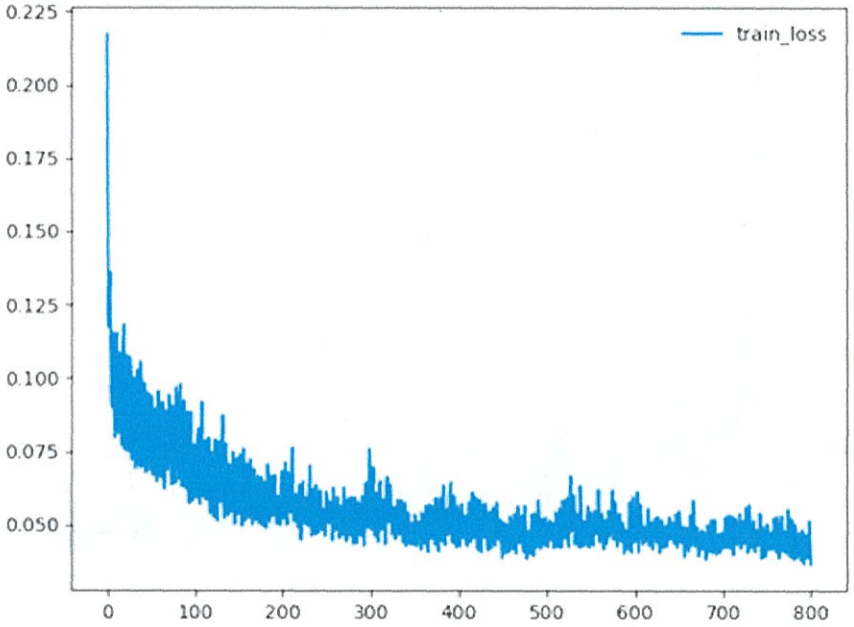


図 5.34. 歩く人データにする LSTM の loss

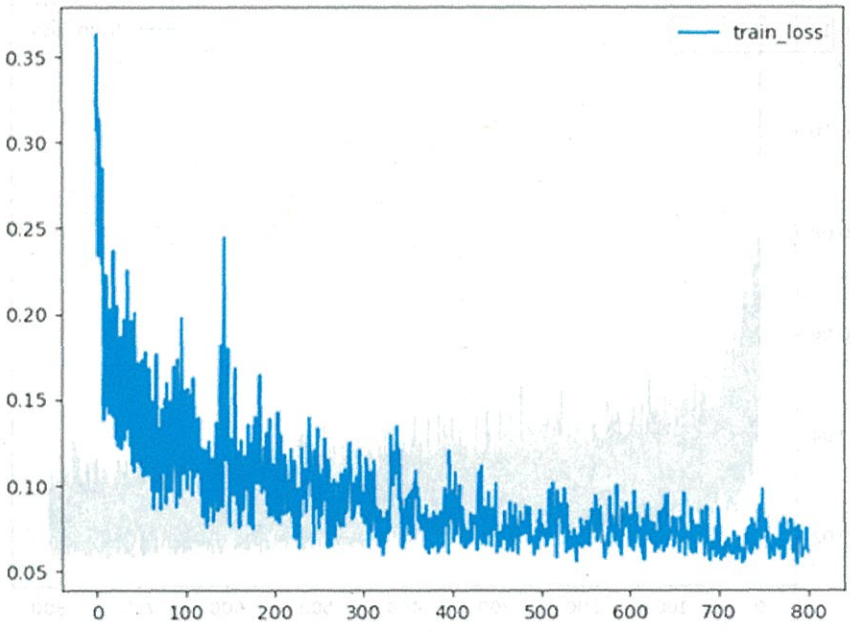


図 5.35. 走る人データにする LSTM の loss

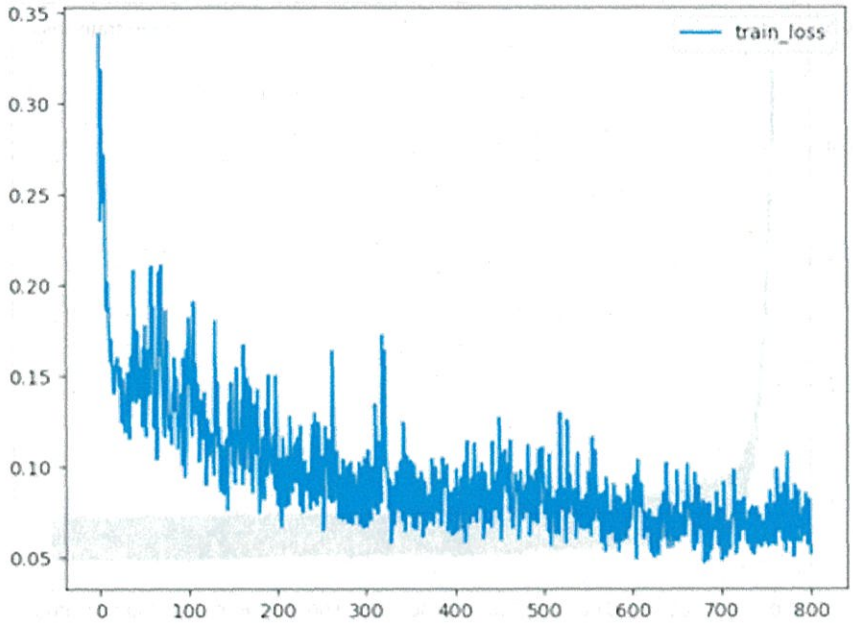


図 5.36. ジャンプする人データにする LSTM の loss

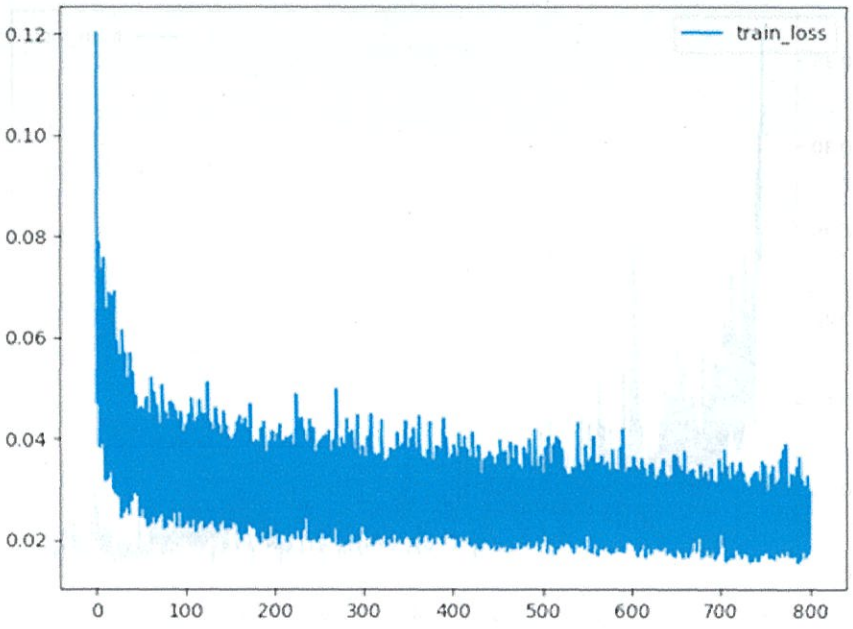


図 5.37. 車椅子利用者データにする LSTM の loss

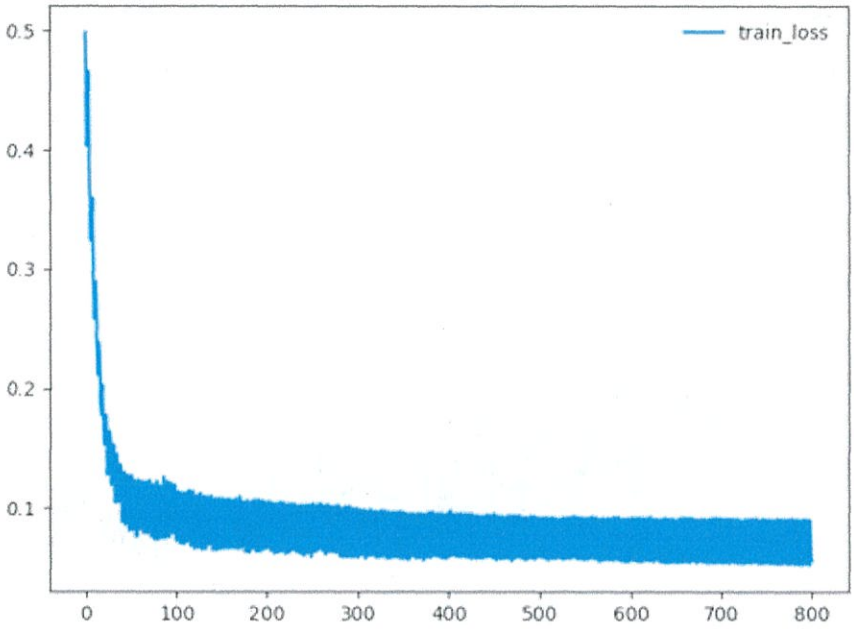


図 5.38. 歩く人データにする MLP の loss

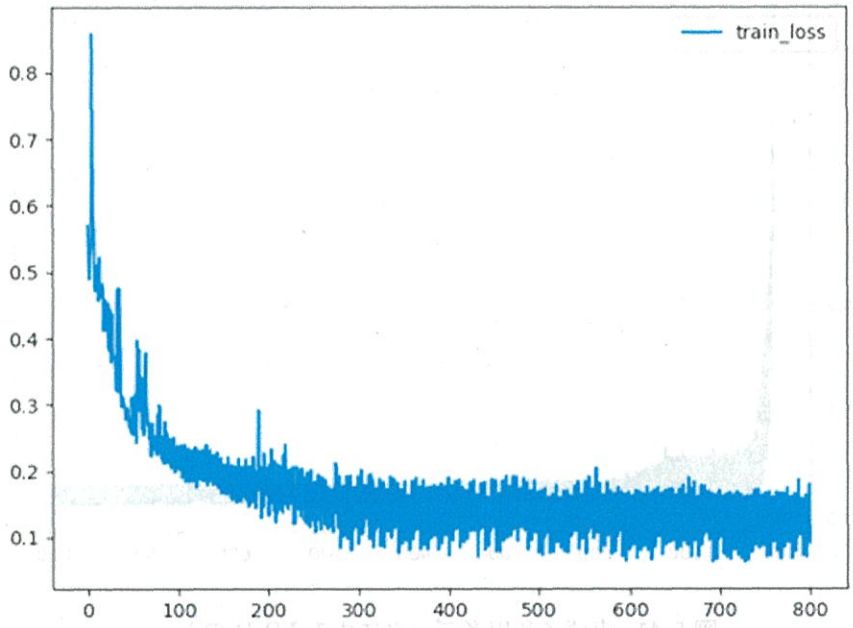


図 5.39. 走る人データにする MLP の loss

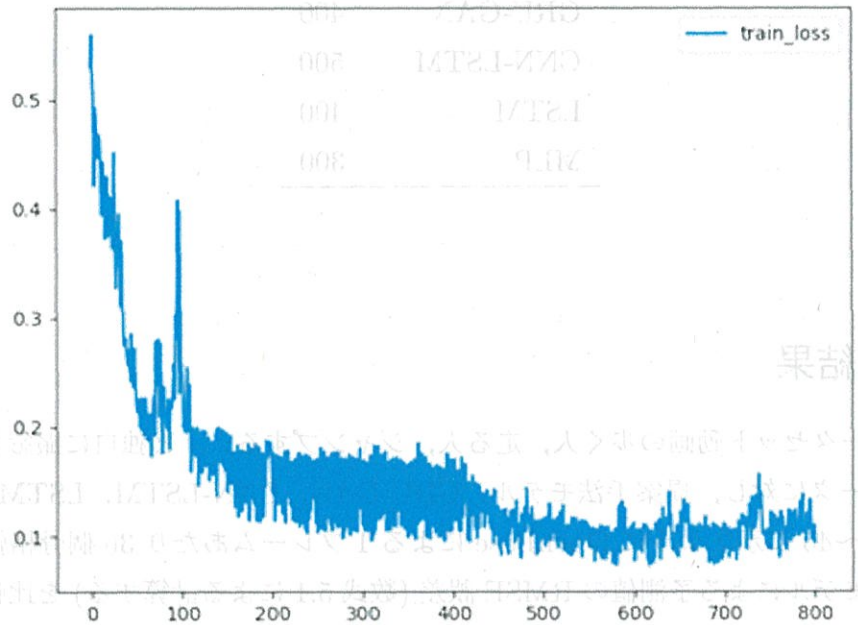


図 5.40. ジャンプする人データにする MLP の loss

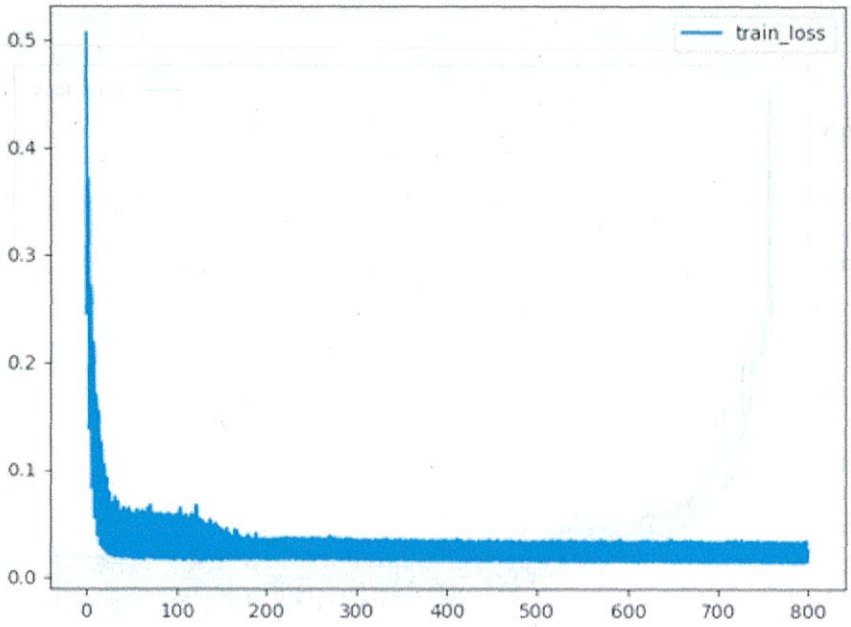


図 5.41. 車椅子利用者データにする MLP の loss

表 5.11. 各モデルの epoch 数の設定

model	epoch
提案手法	400
GRU-GAN	400
CNN-LSTM	500
LSTM	400
MLP	300

5.2　評価結果

MoCap データセット動画の歩く人，走る人，ジャンプする人，と独自に撮影した動画の車椅子利用者データに対し，提案手法モデル，GRU-GAN，CNN-LSTM，LSTM，MLP を用いて，図 5.42〜45 に示すように OpenPose による 1 フレームあたり 36 個骨格情報座標データの測定値とモデルによる予測値の RMSE 誤差 (数式 5.1 による計算する) を比較する．

14 本のテスト動画，合計 1482 フレームに対し，各モデルによる生成した骨格情報座標データの RMSE 誤差の平均値を表 5.12〜14 にまとめた．シミュレーションを簡略化するために，遮蔽が発生した動画の使用ではなく，単純に乱数行列を 3 分割し，3 分の 1，3 分の 2 と全乱

数を連続的に入れ替えてシミュレーションを行った．また，図 5.46〜49 に全乱数の使用において各種類データに対して各モデルによる生成した骨格情報座標データの RMSE 誤差の箱ひげ図を示す．

$$RMSE = \sqrt{\frac{1}{36}\Sigma_{j=1}^{36}\left(d_j - d'_j\right)^2} \qquad (d : \text{測定値}, d' : \text{予測値}) \tag{5.1}$$

表 5.12. 33 %部分遮蔽場合に生成した骨格情報座標データの RMSE の平均値

	Walk	Run	Jump	Wheelchair
提案手法	**1.962**	**7.692**	**3.471**	**15.212**
GRU-GAN	2.142	8.318	3.750	16.295

表 5.13. 67 %部分遮蔽場合に生成した骨格情報座標データの RMSE の平均値

	Walk	Run	Jump	Wheelchair
提案手法	**1.969**	**7.704**	**3.452**	**14.912**
GRU-GAN	2.167	8.271	3.752	16.338

表 5.14. 100 %全身遮蔽場合に生成した骨格情報座標データの RMSE の平均値

	Walk	Run	Jump	Wheelchair
提案手法	1.973	**7.695**	**3.430**	**14.698**
GRU-GAN	2.140	8.306	3.785	16.282
CNN-LSTM	2.743	8.471	3.844	16.624
LSTM	2.070	8.172	4.713	19.385
MLP	**1.955**	9.508	4.422	16.373

図 5.42. 提案手法による生成した歩く人の骨格情報データ

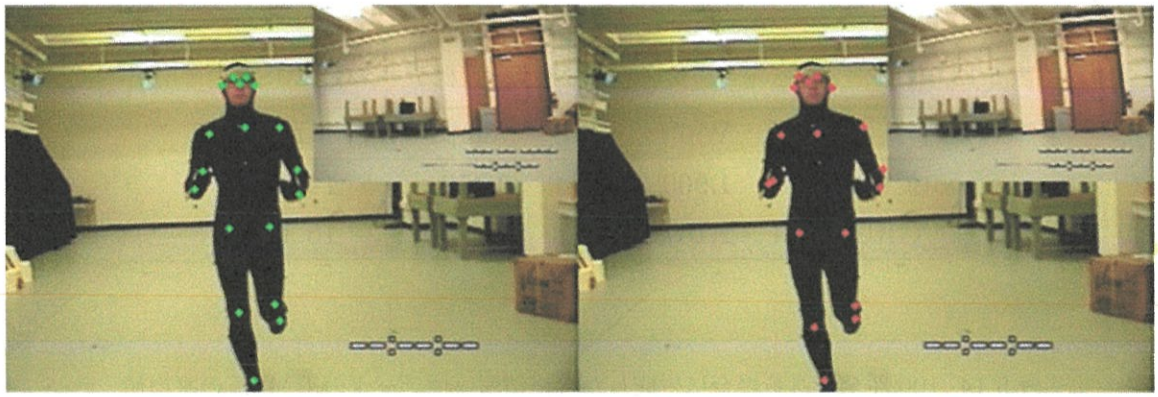


図 5.43. 提案手法による生成した走る人の骨格情報データ

図 5.44. 提案手法による生成したジャンプする人の骨格情報データ

図 5.45. 提案手法による生成した車椅子利用者の骨格情報データ

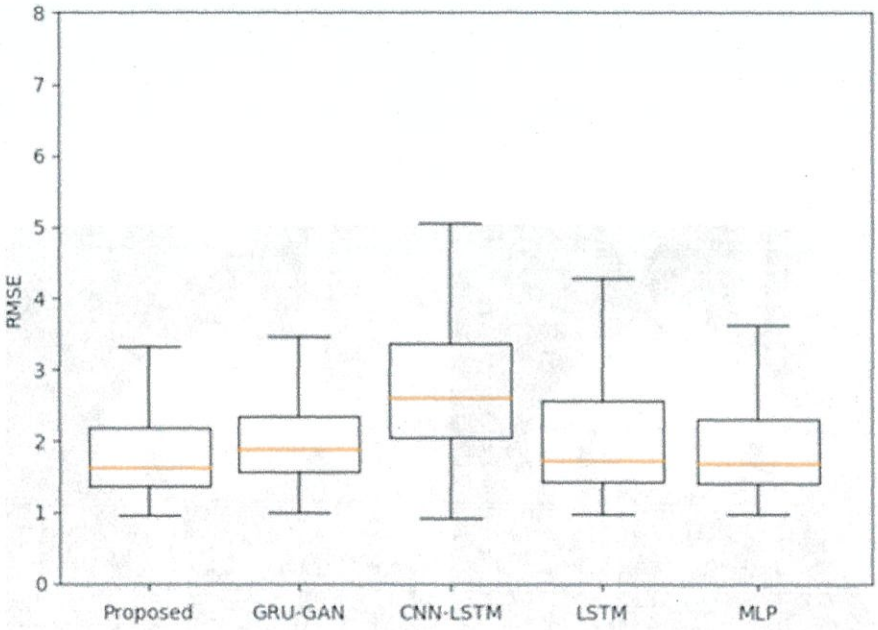


図 5.46. 歩く人データに対する全身遮蔽場合に各モデルの箱ひげ図

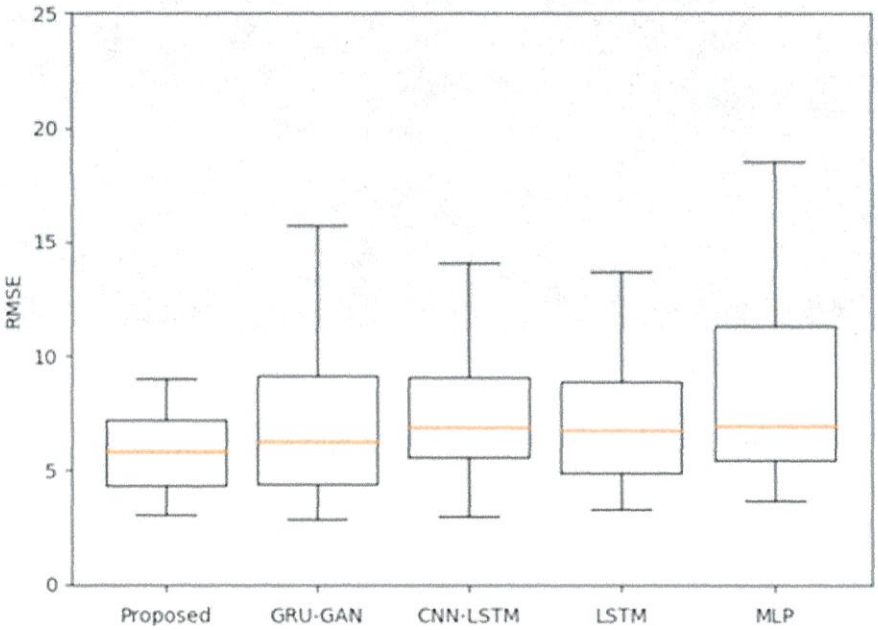


図 5.47. 走る人データに対する全身遮蔽場合に各モデルの箱ひげ図

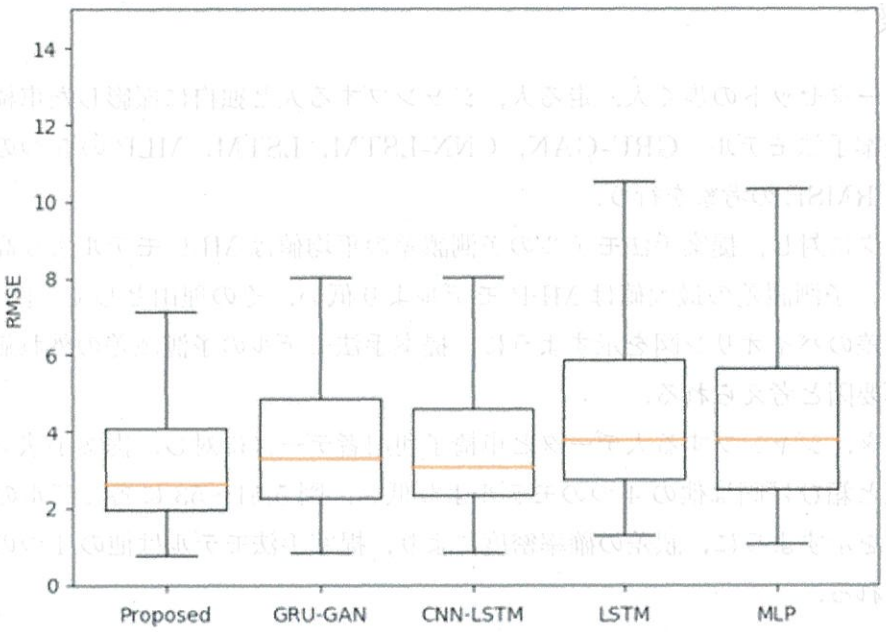


図 5.48. ジャンプする人データに対する全身遮蔽場合に各モデルの箱ひげ図

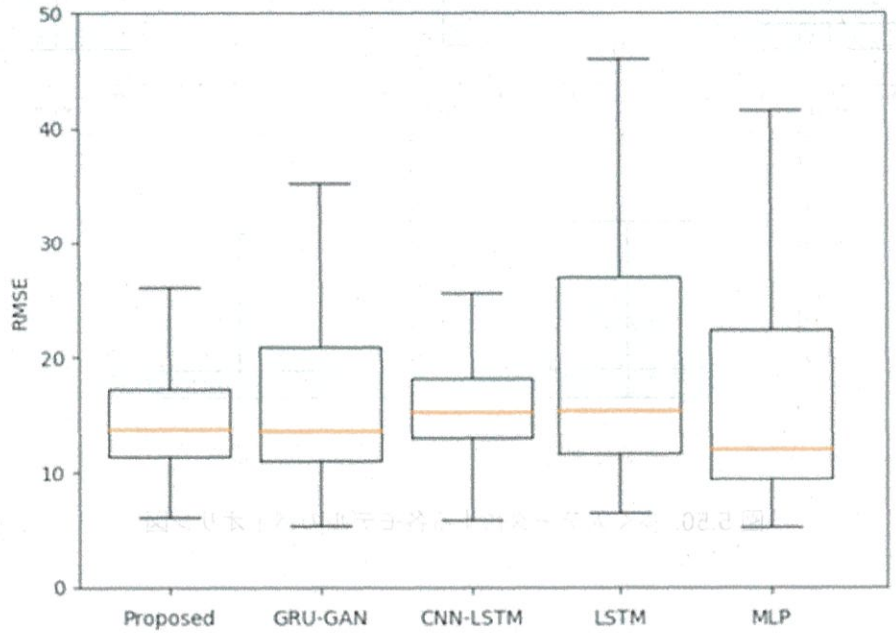


図 5.49. 車椅子利用者データに対する全身遮蔽場合に各モデルの箱ひげ図

5.3　考察

MoCap データセットの歩く人，走る人，ジャンプする人と独自に撮影した車椅子利用者動画に対して提案手法モデル，GRU-GAN，CNN-LSTM，LSTM，MLP の 5 つのモデルを用いた予測誤差 RMSE の考察を行う．

歩く人データに対し，提案手法モデルの予測誤差の平均値は MLP モデルより高いが，箱ひげ図において，予測誤差の最大値は MLP モデルより低い．その理由として，図 5.50 に各モデルの予測誤差のバイオリン図を示すように，提案手法モデルの予測誤差の外れ値は MLP より高いことが要因と考えられる．

走る人データ，ジャンプする人データと車椅子利用者データに対し，提案手法モデルの予測誤差の平均値と箱ひげ図は他の 4 つのモデルより低い．図 5.51～53 に各モデルの予測誤差のバイオリン図を示すように，誤差の確率密度により，提案手法モデルは他の 4 つのモデルより良いと考えられる．

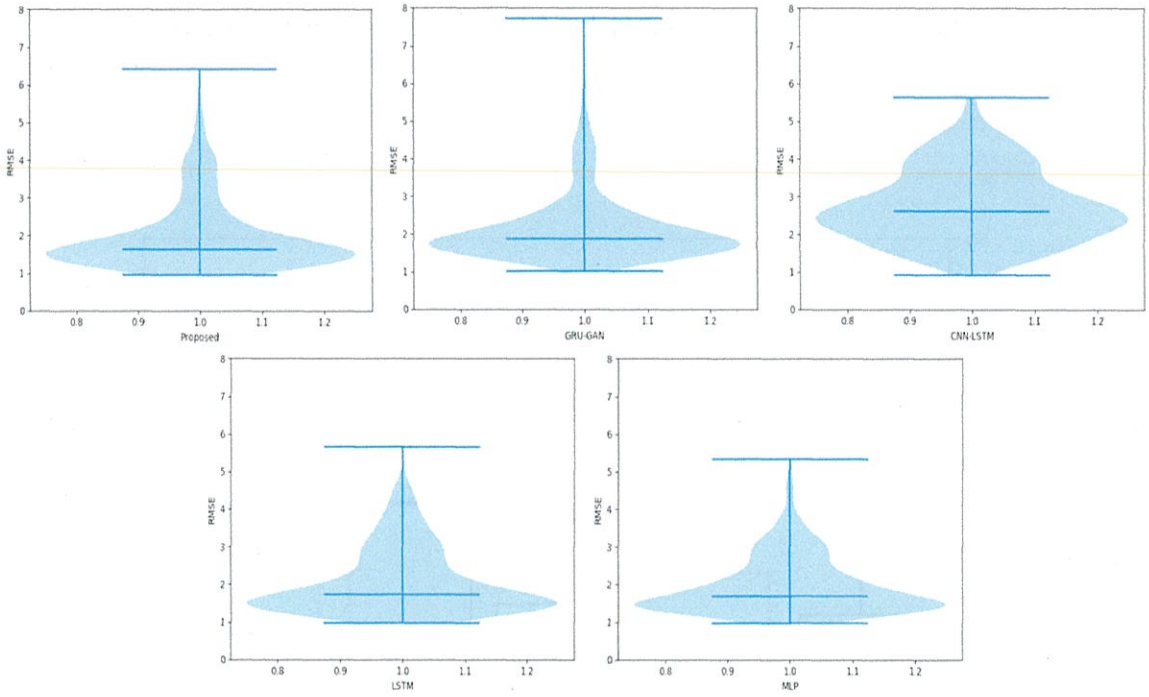


図 5.50. 歩く人データにする各モデルのバイオリン図

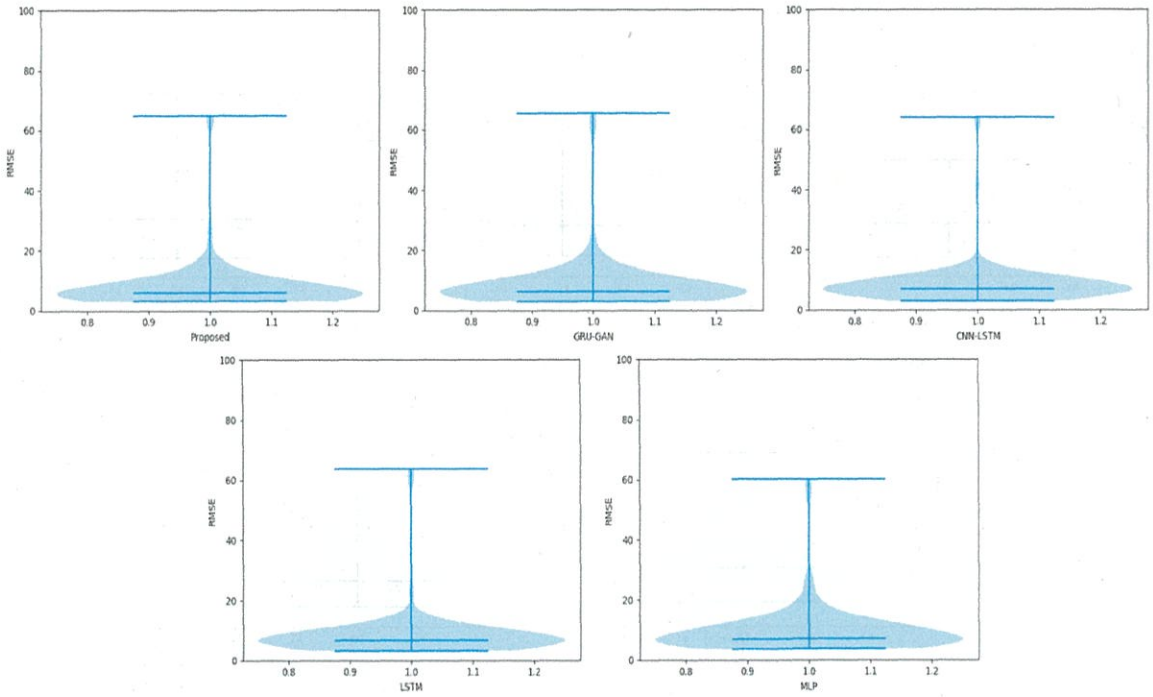


図 5.51. 走る人データにする各モデルのバイオリン図

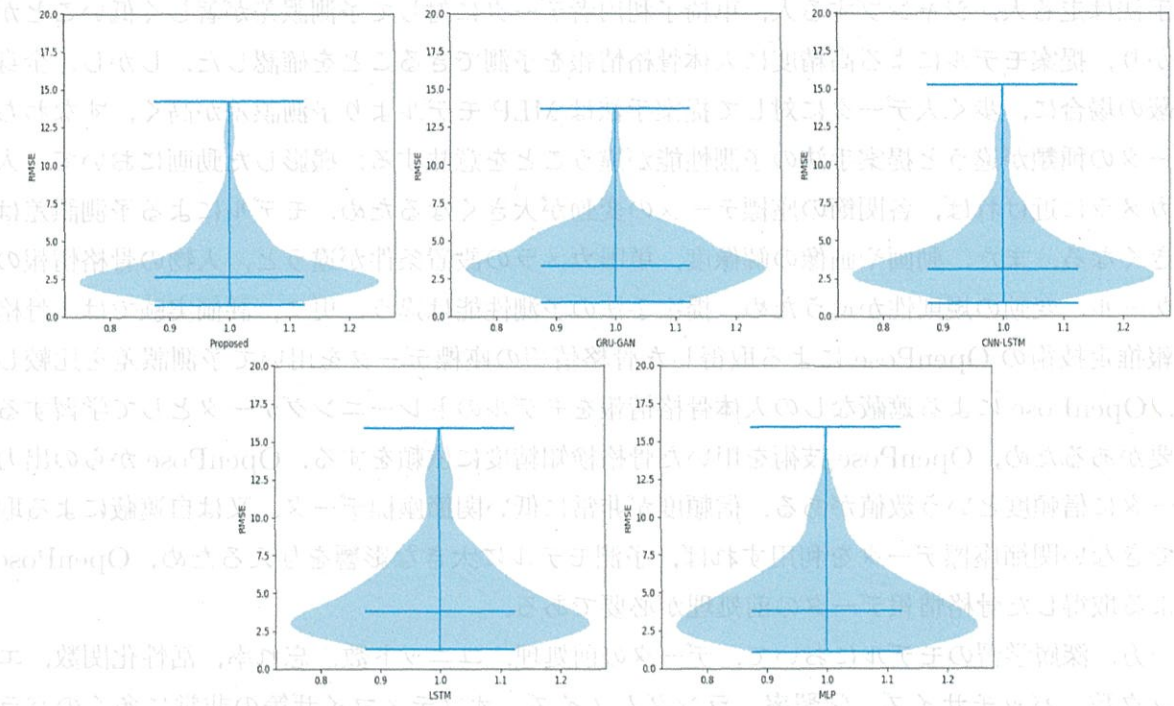


図 5.52. ジャンプする人データにする各モデルのバイオリン図

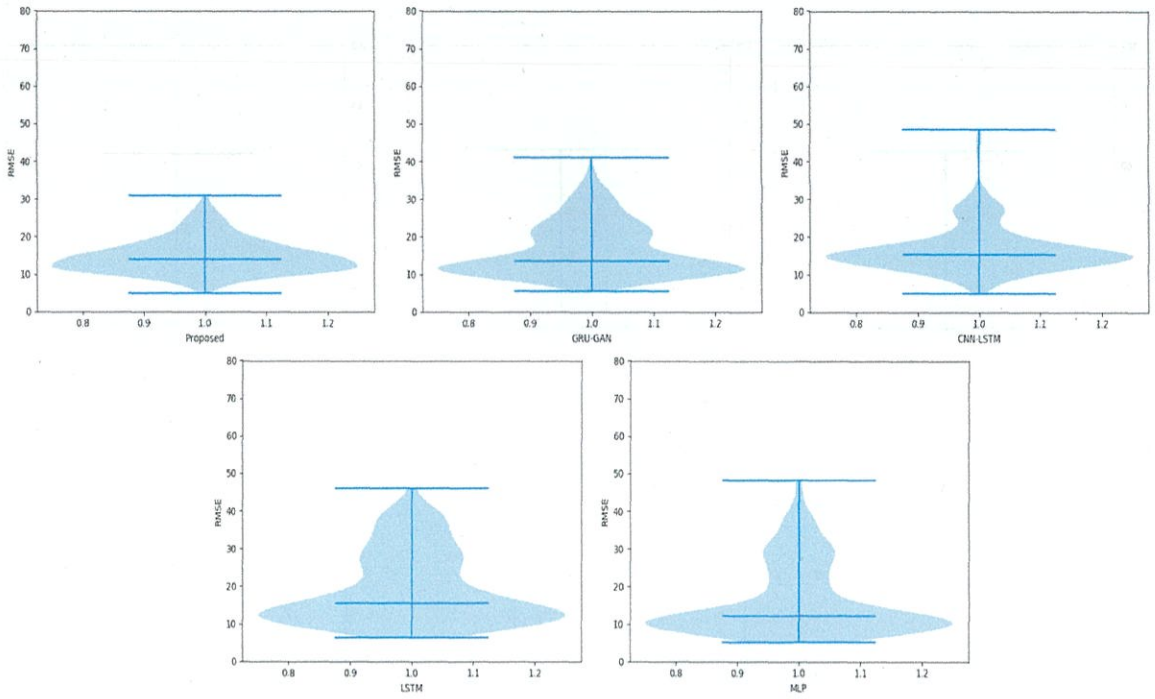


図 5.53. 車椅子利用者データにする各モデルのバイオリン図

総じて，生成された骨格情報座標データの RMSE 平均値の比較 (表 5.12〜14) によって，提案手法は走る人，ジャンプする人，車椅子利用者データに対して予測誤差が著しく低いことが分かり，提案モデルによる高精度に人体骨格情報を予測できることを確認した．しかし，全身遮蔽の場合に，歩く人データに対して提案手法は MLP モデルより予測誤差が高く，すなわちデータの種類が違うと提案手法の予測性能が違うことを意味する．撮影した動画において，人がカメラに近ければ，各関節の座標データの変動が大きくなるため，モデルによる予測誤差は大きくなる．また，動画や画像の解像度，単眼カメラの設置条件が違うと，人物の骨格情報のスケール，変動の規則性が違うため，提案手法の予測性能は違う．更に，評価実験では，骨格情報推定技術の OpenPose による取得した骨格情報の座標データを用いて予測誤差を比較した．OpenPose による遮蔽なしの人体骨格情報をモデルのトレーニングデータとして学習する必要があるため，OpenPose 技術を用いた骨格検知精度に依頼をする．OpenPose からの出力データに信頼度という数値がある．信頼度が非常に低い関節座標データ，又は自遮蔽による取得できない関節座標データを利用すれば，予測モデルに大きな影響を与えるため，OpenPose による取得した骨格情報データの前処理が必要である．

一方，深層学習のモデルにおいて，データの前処理，ユニット数，忘れ率，活性化関数，エポック数，バッチサイズ，学習率，ランダムノイズ，オプティマイザ等の非常に多くのパラメータがあるため，パラメータの設定が違うと提案モデルの予測性能に影響を与える可能性がある．このように，提案モデルを最適化することが困難であると考えられ，本研究ではできる

限り通常の論文またはAI技術ウェブサイトで使われているパラメータを設定した．

最後に，本提案手法においてGANを作成するために使用した学習データのフレーム数は1200左右である．画像認識において，特にGANを用いたノイズから画像を生成する研究で，学習データは数万から数百万のデータ数を要することが多い．本提案手法において，2Dの画像データを利用せず，1Dの座標データを使用し，骨格情報予測のGANネットワークを作成したが，学習データの量と種類を増やすことで，予測性能は更に上がると考えられる．

以上の評価結果より，本手法は時間的空間的制約条件を考慮し，提案手法により，走る人，ジャンプする人，車椅子利用者に対する人体骨格情報の予測誤差が著しく低いことを確認した．しかし，人物の動作種類によって，予測性能が違うため，ニューラルネットワーク構造とパラメータ設定の改善の余地がある．更に多くの種類の運動データや，Kinect，DeepPose，VisionPoseなどの他の骨格情報推定技術を用いた予測性能のシミュレーションを行う必要があると言える．

第 6 章

結論

本論文では，骨格情報推定技術 OpenPose による取得した関節座標データに対し，GAN を用いた遮蔽された部位の座標データを再構成する手法を提案した．GAN において，時間的空間的生成ネットワークと識別ネットワークを構築した．生成ネットワークは LSTM 層，全結合層，融合層による構成され，識別ネットワークは MLP 層による構成された．評価実験では，提案手法モデルの予測誤差を GRU-GAN，CNN-LSTM，LSTM，MLP の 4 つのモデルと比較して評価した．公開データセット MoCap と独自に撮影した車椅子利用者の動画を用いたシミュレーションを行い，走る人，ジャンプする人，車椅子利用者に対し，提案手法モデルの優位性を確認した．

今後の課題として，Kinect 700，UCF 101 などの他の公開データセット，ダンス，バスケットボールなどの他の種類の動作データのシミュレーションを行う．本研究では，MoCap データセットの 3 種類 24 本動画を利用したが，動画においてカメラに正対する測定対象のみを含んでいる．モデルの汎用性を向上するために，監視カメラに正対する人に限らず，横向きに走行したり，後ろを向いて走行している人物の骨格情報を高精度に予測することが必要となる．また，車椅子利用者のみならず，運転者，障がい者，酔客，歩きスマホをする人等の特定の姿勢を維持する人物の骨格情報を用いた危険予測を可能とする．

一方，遮蔽された部位の服装，手足，動作，表情どの復元については，LSTM に限らず，他のモデルを併用し，生成ネットワークの構造を改善することが必要である．例えば，空間的な CNN ネットワークと時間的 LSTM ネットワークを並行的に構築し，CNN による服装，手足などの画像データを学習し，LSTM による動作，部位追跡などの時系列データを学習し，遮蔽された部位の全ての情報を復元できると考えられる．また，LSTM を用いた機械翻訳に関する研究があり，本論文では時系列骨格情報データを利用したが，提案したモデルを改善し，自然言語分野への適用も期待できる．例えば，文字を音声入力している時，通信不良又は環境雑音により，欠損音声，認識できない音声に対し，GAN ネットワークを用いて復元できると考えられる．

謝辞

本研究を進めるにあたり，細やかな御指導，御鞭撻をいただいた次世代情報ネットワーク研究室 朝香 卓也 教授，西辻 崇 助教に深くお礼申し上げます．また，本研究に関して議論して頂き朝香研究室の皆様に深謝し，今後益々のご発展をお祈り申し上げます．最後に，研究活動だけでなく日々の生活においても大変お世話になった朝香研究室の先輩、同輩、後輩諸氏に心より感謝いたします．

参考文献

[1] Ning C., Yuqing C., Haiqiang L., Lingtao H., and Hongyan Z., "Human Pose Recognition Based on Skeleton Fusion from Multiple Kinects" , 2018 37th Chinese Control Conference (CCC), 2018.

[2] Junwei L., Guoliang L., Guohui T., Xianglai Z., and Ziren W., "Distributed RGBD Camera Network for 3D Human Pose Estimation and Action Recognition" , 2018 21st International Conference on Information Fusion (FUSION), 2018.

[3] Ziren W., Guoliang L., and Guohui T., "Human Skeleton Tracking Using Information Weighted Consensus Filter in Distributed Camera Networks" , 2017 Chinese Automation Congress (CAC), 2017.

[4] Yuanjie W., Lei G., Simon H., and Robert W. L., "Towards Robust 3D Skeleton Tracking Using Data Fusion from Multiple Depth Sensors" , 2018 10th International Conference on Virtual Worlds and Games for Serious Applications (VS-Games), 2018.

[5] Hubert P. H. S., Edmond S. L. H., Yang J., and Shu T., "Real-Time Posture Reconstruction for Microsoft Kinect" , IEEE Transactions on Cybernetics, vol.43, no.5, pp.1357-1369, 2013.

[6] Tanikawa U., Kawanishi Y., Deguchi D., Ide I., Murase H., and Kawai R., "Wheelchair-User Detection Combined with Parts-Based Tracking" , 12th International Joint Conference on Computer Vision, Imaging and Computer GraphicsTheory and Applications (VISIGRAPP), vol.5, pp.165-172, 2017.

[7] Yuuki H., Yasutoshi M., and Hiroyuki S., "Computational Foresight: Forecasting Human Body Motion in Real-Time for Reducing Delays in Interactive System" , 2017 ACM International Conference on Interactive Surfaces and Spaces, pp.312-317, 2017.

[8] Rosenblatt F., "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain" , Psychological Review, vol.65, No.6, 1958.

[9] Songyang Z., Yang Y., Jun X., Xiaoming L., Yi Y., Di X., and Yueting Z., "Fusing Geometric Features for Skeleton-Based Action Recognition Using Multilayer LSTM

Networks" , IEEE Transactions on Multimedia, vol.20, no.9, pp.2330-2343, 2018.

[10] Inwoong L., Doyoung K., Seoungyoon K., and Sanghoon L., "Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks" , IEEE International Conference on Computer Vision (ICCV), 2017.

[11] Tae-Young K., and Sung-Bae C., "Particle Swarm Optimization-Based CNN-LSTM Networks for Forecasting Energy Consumption" , IEEE Congress on Evolutionary Computation (CEC), 2019.

[12] Zhanhong H., Junhao Z., Hong-Ning D., and Hao W., "Gold Price Forecast Based on LSTM-CNN Model" , IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), 2019.

[13] Emad B., John K., and Zicheng L., "HP-GAN: Probabilistic 3D Human Motion Prediction via GAN" , IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018.

[14] Cho K., Van Merrinboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., and Bengio Y., "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation" , arXiv preprint arXiv:1406.1078, 2014.

[15] Tero K., Samuli L., and Timo A., "A Style-Based Generator Architecture for Generative Adversarial Networks" , IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.4401-4410, 2019.

[16] Daichi H., Wataru S., and Keiji Y., "Unseen Food Creation by Mixing Existing Food Images with Conditional StyleGAN" , 5th International Workshop on Multimedia Assisted Dietary Management, pp.19-24, 2019.

[17] Bhler M., Park S., De Mello S., Zhang X., and Hilliges O., "Content-Consistent Generation of Realistic Eyes with Style" , arXiv preprint arXiv:1911.03346, 2019.

[18] Fu C., Hu Y., Wu X., Wang G., Zhang Q., and He R., "High Fidelity Face Manipulation with Extreme Pose and Expression" , arXiv preprint arXiv:1903.12003, 2019.

[19] Mengyu C., You X., Laura L., and Nils T., "Temporally Coherent GANs for Video Super-Resolution (TecoGAN)" , arXiv preprint arXiv:1811.09393, 2018.

[20] Weiss S., Chu M., Thuerey N., and Westermann R., "Volumetric Isosurface Rendering with Deep Learning-Based Super-Resolution", arXiv preprint arXiv:1906.06520, 2019.

[21] Pourreza R., Ghodrati A., and Habibian A., "Recognizing Compressed Videos: Challenges and Promises" , IEEE International Conference on Computer Vision Work-

shops, 2019.

[22] Thu N., Chuan L., Lucas T., Christian R., and Yong-Liang Y., "HoloGAN: Unsupervised Learning of 3D Representations from Natural Images" , arXiv preprint arXiv:1904.01326, 2019.

[23] Noguchi A., and Harada T., "RGBD-GAN: Unsupervised 3D Representation Learning From Natural Image Datasets via RGBD Image Synthesis" , arXiv preprint arXiv:1909.12573, 2019.

[24] Kato H., and Harada T., "Self-Supervised Learning of 3D Objects from Natural Images" , arXiv preprint arXiv:1911.08850, 2019.

[25] Jun-Yan Z., Taesung P., Phillip I., and Alexei A. E., "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks" , The European Conference on Computer Vision (ECCV), pp.184-199, 2018.

[26] Almahairi A., Rajeswar S., Sordoni A., Bachman P., and Courville A, "Augmented CycleGAN: Learning Many-to-Many Mappings from Unpaired Data", arXiv preprint arXiv:1802.10151, 2018.

[27] Lu Y., Tai Y. W., and Tang C. K., "Attribute-Guided Face Generation Using Conditional CycleGAN" , The European Conference on Computer Vision (ECCV), pp.282-297, 2018.

[28] Hosseini-Asl E., Zhou Y., Xiong C., and Socher R., "A Multi-Discriminator CycleGAN for Unsupervised Non-Parallel Speech Domain Adaptation" , arXiv preprint arXiv:1804.00522, 2018.

[29] Engin D., Gen A., and Kemal E. H, "Cycle-Dehaze: Enhanced CycleGAN for Single Image Dehazing" , IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp.825-833, 2018.

[30] Chang B., Zhang Q., Pan S., and Meng L., "Generating Handwritten Chinese Characters Using CycleGAN" , In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp.199-207, 2018.

[31] Yi L., Ying X., and Shao-bin L., "2-D Human Pose Estimation from Images Based on Deep Learning: A Review" , 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2018.

[32] Saeid V., and Maria A. A., "Deep 3D Human Pose Estimation under Partial Body Presence" , 2018 25th IEEE International Conference on Image Processing (ICIP), 2018.

[33] Fei G., Yifeng H., and Ling G., "RGB-D Camera Pose Estimation Using Deep Neural Network", 2017 IEEE Global Conference on Signal and Information Processing

(GlobalSIP), 2017.

[34] Zhe C., Tomas S., Shih-En W., and Yaser S., "Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields" , IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[35] Adrian B., and Georgios T., "Human Pose Estimation via Convolutional Part Heatmap Regression" , The 14th European Conference on Computer Vision (ECCV), 2016.

[36] Shih-En W., Varun R., Takeo K., and Yaser S., "Convolutional Pose Machines" , IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[37] Sen Q., Yilin W., and Jian L., "Real-Time Human Gesture Grading Based on OpenPose" , 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2017.

[38] Hossein F., Amin M., Hawre H., and Rainer H., "Swim Stroke Analytic: Front Crawl Pulling Pose Classification" , 2018 25th IEEE International Conference on Image Processing (ICIP), 2018.

[39] Paschalis P., Iason O., and Antonis A., "Using a Single RGB Frame for Real Time 3D Hand Pose Estimation in the Wild" , 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018.

[40] Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., and Bengio Y., "Generative Adversarial Nets" , In Advances in Neural Information Processing Systems, pp.2672-2680, 2014.

[41] Hochreiter S., and Jrgen S., "Long Short-Term Memory", Neural Computation 9(8), pp.1735-1780, 1997.

[42] Jianlong F., Heliang Z., and Tao M., "Look Closer to See Better: Recurrent Attention Convolutional Neural Network for Fine-Grained Image Recognition", IEEE Conference on Computer Vision and Pattern Recognition, pp.4438-4446, 2017.

[43] Fei W., Mengqing J., Chen Q., Shuo Y., Cheng L., Honggang Z., Xiaogang W., and Xiaoou T., "Residual Attention Network for Image Classification", IEEE Conference on Computer Vision and Pattern Recognition, pp.3156-3164, 2017.

[44] Liang-Chieh C., Yi Y., Jiang W., Wei X., and Alan L. Y., "Attention to Scale: Scale-Aware Semantic Image Segmentation", IEEE Conference on Computer Vision and Pattern Recognition, pp.3640-3649, 2016.

[45] Kelvin X., Jimmy B., Ryan K., Kyunghyun C., Aaron C., Ruslan S., Rich Z., and Yoshua B., "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", In International Conference on Machine Learning, pp.2048-2057, 2015.

[46] Kaiming H., Georgia G., Piotr D., and Ross G., "Mask R-CNN", IEEE International Conference on Computer Vision, pp.2961-2969, 2017.

付録

Listing 6.1. 提案手法の生成ネットワークのソースコード断片

```
input_g = Input(shape=(1, obs_in))
lstm_1 = LSTM(256, dropout=0.2, name='lstm_1', return_sequences=False)(
    input_g)
advanced_activations.LeakyReLU(alpha=0.2)
attention_probs = Dense(256, activation='sigmoid', name='attention_1')(
    lstm_1)
attention_out = multiply([lstm_1, attention_probs])
dense_out = Dense(256, name='dense_1')(attention_out)
advanced_activations.LeakyReLU(alpha=0.2)
output_g = Dense(obs_out, activation='sigmoid', name='dense_2')(dense_out)
model_G = Model(inputs=input_g, outputs=output_g)
model_G.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
```

Listing 6.2. 提案手法の識別ネットワークのソースコード断片

```
model_D = Sequential()
model_D.add(Dense(64, input_dim=obs_out))
advanced_activations.LeakyReLU(alpha=0.2)
model_D.add(Dense(16))
advanced_activations.LeakyReLU(alpha=0.2)
model_D.add(Dense(4))
advanced_activations.LeakyReLU(alpha=0.2)
model_D.add(Dense(1, activation='sigmoid'))
model_D.compile(loss='binary_crossentropy', optimizer='RMSprop', metrics
    =['accuracy'])
```

Listing 6.3. 提案手法の GAN のソースコード断片

```
model_D.trainable = False
gan_input = keras.Input(shape=(1, obs_in))
```

```
gan_output = model_D(model_G(gan_input))
gan = keras.models.Model(gan_input, gan_output)
gan_optimizer = keras.optimizers.RMSprop(lr=0.0001)
gan.compile(optimizer=gan_optimizer, loss='binary_crossentropy')
```

Listing 6.4. GRU-GAN の生成ネットワークのソースコード断片

```
model_G = Sequential()
model_G.add(GRU(256, dropout=0.2, input_shape=(1, obs_in)))
model_G.add(Dense(256))
advanced_activations.LeakyReLU(alpha=0.2)
model_G.add(Dense(obs_out))
advanced_activations.LeakyReLU(alpha=0.2)
model_G.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
```

Listing 6.5. GRU-GAN の識別ネットワークのソースコード断片

```
model_D = Sequential()
model_D.add(Dense(64, input_dim=obs_out))
advanced_activations.LeakyReLU(alpha=0.2)
model_D.add(Dense(16))
advanced_activations.LeakyReLU(alpha=0.2)
model_D.add(Dense(4))
advanced_activations.LeakyReLU(alpha=0.2)
model_D.add(Dense(1, activation='sigmoid'))
model_D.compile(loss='binary_crossentropy', optimizer='RMSprop', metrics
    =['accuracy'])
```

Listing 6.6. GRU-GAN の GAN のソースコード断片

```
model_D.trainable = False
gan_input = keras.Input(shape=(1, obs_in))
gan_output = model_D(model_G(gan_input))
gan = keras.models.Model(gan_input, gan_output)
gan_optimizer = keras.optimizers.RMSprop(lr=0.0001)
gan.compile(optimizer=gan_optimizer, loss='binary_crossentropy')
```

Listing 6.7. CNN-LSTM のソースコード断片

```
input_cl = Input(shape=(1, obs_in))
conv1d_1_1 = Conv1D(filters=256, kernel_size=1, name='conv1d_1_1')(
    input_cl)
```

```
3 advanced_activations.LeakyReLU(alpha=0.2)
4 maxpooling_1 = MaxPooling1D(pool_size=2, padding='SAME', name='
    maxpooling_1')(conv1d_1_1)
5 conv1d_2_1 = Conv1D(filters=128, kernel_size=1, name='conv1d_2_1')(
    maxpooling_1)
6 advanced_activations.LeakyReLU(alpha=0.2)
7 maxpooling_2 = MaxPooling1D(pool_size=2, padding='SAME', name='
    maxpooling_2')(conv1d_2_1)
8 conv1d_3_1 = Conv1D(filters=64, kernel_size=1, name='conv1d_3_1')(
    maxpooling_2)
9 advanced_activations.LeakyReLU(alpha=0.2)
10 lstm_1 = LSTM(256, dropout=0.2, name='lstm_1', return_sequences=False)(
    conv1d_3_1)
11 advanced_activations.LeakyReLU(alpha=0.2)
12 attention_probs = Dense(256, activation='sigmoid', name='attention_1')(
    lstm_1)
13 attention_out = multiply([lstm_1, attention_probs])
14 output_cl = Dense(obs_out, activation='sigmoid', name='dense_1')(
    attention_out)
15 model_CL = Model(inputs=input_cl, outputs=output_cl)
16 model_CL.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
```

Listing 6.8. LSTM のソースコード断片

```
1 model_L = Sequential()
2 model_L.add(LSTM(256, dropout=0.2, input_shape=(1, obs_in)))
3 model_L.add(Dense(256))
4 advanced_activations.LeakyReLU(alpha=0.2)
5 model_L.add(Dense(obs_out))
6 advanced_activations.LeakyReLU(alpha=0.2)
7 model_L.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
```

Listing 6.9. MLP のソースコード断片

```
1 model_M = Sequential()
2 model_M.add(Dense(64, input_dim=obs_in))
3 advanced_activations.LeakyReLU(alpha=0.2)
4 model_M.add(Dense(16))
5 advanced_activations.LeakyReLU(alpha=0.2)
6 model_M.add(Dense(4))
7 advanced_activations.LeakyReLU(alpha=0.2)
```

```
8 model_M.add(Dense(obs_out))
9 advanced_activations.LeakyReLU(alpha=0.2)
10 model_M.compile(loss='mae', optimizer='adam', metrics=['accuracy'])
```